

UNITED STATES AIR FORCE RESEARCH LABORATORY

Automating Maintenance Instructions Study: Procedure Planning Technologies

Stephen Deutsch
Richard Pew
Robert Granville
Bruce Roberts
Alice Mulvehill
Michael Cramer

BBN Systems and Technologies
10 Moulton Street
Cambridge MA 02138

February 1998

Final Report for the Period June 1997 to January 1998

19980817 049

Approved for public release; distribution is unlimited.

Human Effectiveness Directorate
Crew Survivability and Logistics Division
Sustainment Logistics Branch
2610 Seventh Street
Wright-Patterson AFB OH 45433-7901

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner, licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

Please do not request copies of this paper from the Air Force Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield VA 22161

Federal Government agencies and their contractors registered with Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
8725 John J. Kingman Rd., STE 0944
Ft Belvoir VA 22060-6218

DISCLAIMER

This Technical Paper is published as received and has not been edited by the Air Force Research Laboratory, Human Effectiveness Directorate.


TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TP-1998-0004

This paper has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical paper has been reviewed and is approved for publication.

FOR THE COMMANDER

For  Lt Col
THOMAS J. MOORE, Chief
Crew Survivability and Logistics Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE February 1998		3. REPORT TYPE AND DATES COVERED Final - June 1997 - January 1998
4. TITLE AND SUBTITLE Automating Maintenance Instructions Study: Procedure Planning Technologies			5. FUNDING NUMBERS C - F41624-97-D-5023 PE - 62205F PR - 1710 TA - D0 WU - 04	
6. AUTHOR(S) Stephan Deutsch, Richard Pew, Robert Granville, Bruce Roberts, Alice Mulvehill, Michael Cramer				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) BBN Systems and Technologies 10 Moulton Street Cambridge MA 02138			8. PERFORMING ORGANIZATION REPORT NUMBER BBN No. 8231	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory, Human Effectiveness Directorate Crew Survivability and Logistics Division Air Force Materiel Command Sustainment Logistics Branch Wright-Patterson AFB OH 45433-7604			10. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-HE-WP-TP-1998-0004	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This report describes several approaches to automating the processes surrounding original creation of maintenance instructions, or technical orders. The report emphasizes techniques rooted in linguistics, particularly computational linguistics, that seem to hold promise for automating the derivation of maintenance procedures conveyed as text. Two varieties are highlighted: case-based planning, which identifies similar procedures from existing technical manuals to define new procedures, and generative planning, which operates from first principles. Other elements of a complete set of tools for automation support are also described. These include natural language understanding, human form modeling, and computer-aided design integration. Several case studies are used to illustrate the potential of suggested approaches to the automation problem.				
14. SUBJECT TERMS case-based planning, generative planning, computational linguistics, maintenance technical manuals			15. NUMBER OF PAGES 54	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

The work on the Automating Maintenance Instructions Study was conducted under Delivery Order 8 of the Logistics Technology Research Support (LTRS) program administered under U. S. Air Force Contract Number F41624-9T-D-5002.

The authors wish to thank Edward Boyle of Air Force Research Laboratory for his enthusiastic support and constructive criticism of this research effort.

Summary

This study of Automating Maintenance Instructions (AMI) focuses on the interface between the geometry of the device and the verbal description of the maintenance actions required for the human maintainer (currently Technical Orders.) The interface issues are discussed in the context of requirements for geometric models and for the language (text) generation needed to accurately describe these maintenance actions. This report is organized into six main sections, two case studies, recommendations, a glossary of terms, and references. First we discuss the implications of object geometry on maintenance modeling and argue for the consideration of human task activities as an essential component of maintenance procedures planning and instructions. Then we introduce the language generation issues, including distinctions between state-space, kinematic, dynamic, and process control terms. We describe the lexical semantics that is necessary for the generation of precise and accurate verbal instructions. Since instructions will be executed sequentially, an important element of the instruction is specific information with respect to its completion or culmination, and culminating conditions are discussed in detail. The actual text of an instruction is created through processes of text generation and planning. The method by which the same planning process can be extended to include the consideration of a visual presentation of information as well is discussed, and the careful coordination that this would require. We then present the case studies involving a task where the presence of the human maintainer fixes a task ordering that is not determined solely from the geometry data. The animation study addresses collision detection and access requirements over the geometry. The language study looks at the same example from the sentence generation perspective and focuses on lexical choice and precise object description. Finally, we summarize our AMI recommendations.

Table of Contents

1. INTRODUCTION	1
2. AMI INPUT SIDE CAPABILITIES AND SUPPORTING TECHNOLOGIES	10
2.1 TECHNICAL ORDER PROCEDURE REPRESENTATION	12
2.1.1 <i>Maintenance Procedure Representation</i>	13
2.1.2 <i>OMAR Procedure Representation</i>	14
2.1.3 <i>Simulating Procedure Execution</i>	17
2.1.4 <i>One Procedure Representation or Several</i>	19
2.2 LINGUISTIC ANALYSIS AND LANGUAGE UNDERSTANDING FOR AMI	21
2.2.1 <i>Automatic Acquisition of Procedures</i>	22
2.2.2 <i>Text Analysis for Knowledge Acquisition</i>	24
2.2.3 <i>Semantic Parsing for Generative Planning</i>	28
2.2.4 <i>Semantic Parsing for Case Based Reasoning</i>	29
2.2.5 <i>Concluding Remarks on Linguistic Analysis and Language Understanding</i>	29
2.3 PLANNING TECHNOLOGY FOR AMI	30
2.3.1 <i>Using Generative Planning to Promote Automation</i>	31
2.3.1.1 Removal	32
2.3.1.2 Installation	34
2.3.1.3 Authoring as Plan Expansion	34
2.3.2 <i>Using Case-based Planning to Promote Automation</i>	36
2.3.2.1 Related Case Based Reasoning Research and Systems	37
2.3.2.2 A CBR Experiment	38
2.3.2.3 Comparing Plan Components Versus Entire Plans	39
2.3.2.4 Suggestions for CBR Usage	40
2.3.3 <i>Combining CBR and Generative Planners</i>	41
3. CONCLUSIONS AND RECOMMENDATIONS	43
4. GLOSSARY	44
5. REFERENCES	45

Figures

FIGURE 1. AUTOMATION FOR MAINTENANCE PROCEDURE AUTHORIZING	12
FIGURE 2. PROCEDURE REPRESENTATION AS A CENTRAL AMI CAPABILITY	13
FIGURE 3. F-16 FUEL SYSTEM PROCEDURE 2-14-1 (PAGE 1).....	17
FIGURE 4. PROCEDURE BROWSER VIEW	18
FIGURE 5. MAINTENANCE PROCEDURE TRACE	19
FIGURE 6. MAINTENANCE PROCEDURE TIMELINE.....	21
FIGURE 7. ACTIONS IN FUEL TANK PRESSURIZATION VALVE PROCEDURES	22
FIGURE 8. OBJECTS IN FUEL TANK PRESSURIZATION VALVE PROCEDURES.....	23
FIGURE 9. SYNTACTIC PARSE OF "REMOVE ACCESS PANEL 3405."	25
FIGURE 10. SYNTACTIC PARSE OF "DISCONNECT PRESSURE SENSE TUBE."	25
FIGURE 11. SYNTACTIC PARSE OF "PURGE VENT TANK."	26
FIGURE 12. SYNTACTIC PARSE OF "DISCARD PACKING."	26
FIGURE 13. SEMANTIC PARSE OF "REMOVE ACCESS PANEL 3405."	27
FIGURE 14. SEMANTIC RULES TO PARSE "REMOVE ACCESS PANEL 3405."	27
FIGURE 15. SEMANTIC HIERARCHY FOR T.O. PROCEDURE	27
FIGURE 16. PAIR-WISE COMPARISON OF FUEL TANK VENT AND PRESSURIZATION VALVE MAINTENANCE PROCEDURE	32
FIGURE 17. BASIC CASE.....	39
FIGURE 18. CASE COMPARISON REPORT	40

Tables

TABLE 1. TECHNOLOGY INNOVATIONS RANKED BY AVERAGE IMPORTANCE (1.00 IS EXTREMELY IMPORTANT; 5.0 IS UNIMPORTANT)	7
TABLE 2. TECHNOLOGY INNOVATIONS RANKED BY AVERAGE IMPORTANCE OF AIR FORCE INVESTMENT (1.00 = EXTREMELY IMPORTANT; 5.0 = UNIMPORTANT).....	8
TABLE 3. TECHNOLOGY INNOVATIONS RANKED BY AVERAGE RISK (1 = VERY RISKY; 5 = NOT RISKY).....	8
TABLE 4. TECHNOLOGY INNOVATIONS RANKED BY AVERAGE ESTIMATED TIME (YEARS) TO ACHIEVE RESULTS.....	9
TABLE 5. CORRELATION AMONG TECHNOLOGY INNOVATION RATINGS	10
TABLE 6. RECOMMENDED AMI CAPABILITIES AND SUPPORTING TECHNOLOGIES.....	43

1. Introduction

Technical Orders (T.O.s), the manuals used by maintenance personnel to guide them in the maintenance and repair of Air Force equipment, have always been costly to develop and have always incurred further significant costs as they are updated through the life-cycle of any given product. Over the years, there have been significant advances made in the Computer Aided Design (CAD) and Computer Aided Engineering (CAE) tools used in the design of Air Force equipment and the promise is consistently made that this will be reflected in reduced costs for developing and maintaining T.O.s. That the process for authoring T.O.s should become part of the design process is a given. Just how or when this will take place is less firmly pinned down. Progress in Product Data Management (PDM) will be very important in this development. As PDM systems make design data more accessible, those data can be made available more readily to the authoring process for T.O.s.

As the system for the authoring of T.O.s becomes a part of the product design environment more savings will be accrued. Virtual manufacturing, the subject of intense research and development, is representative of the new technologies within the design process that can be adapted in the authoring process. One important concern in virtual manufacturing is the access, fit, and assembly of component parts within a higher level assembly. That component parts are accessible and fit together is an integral part of the authoring process and hence, an area that will profit from the work in virtual manufacturing.

Even with this limited background it is clear that the focus of a study on using technology to aid the authors of T.O.s in their tasks should situate the development of an authoring capability within the newly emerging PDM/CAD/CAE framework. The authoring environment should be a component within this larger framework taking advantage of the new technologies that are coming on line. PDM, for example, should be investigated as the means to provide ready access to the engineering data, the search for which currently consumes so much of the author's time (Sanchez, Winning, & Boyle, 1997).

Our approach to this study is based on our expertise in the technology areas of linguistic analysis, language understanding and text generation, automated planning, simulation, and human figure modeling. These are many of the technology areas cited in the Zimmerman, Green, Gunning, Worrall, & Dimock (1993) study. The focus of the study centers the use of newly emerging technologies in the development of authoring tools to support T.O. generation. Just what form the new T.O.s should take and how that form can be expected to improve the performance of maintenance personnel was outside the scope of the study.

A subject of special concern in the study was that of leverage: How can scarce research dollars be used in a manner that will have the most impact on the problem at hand—product life-cycle costs in the authoring and updating of T.O.s? Building better PDM systems and the new research in virtual manufacturing will certainly have an impact on the authoring of T.O.s, but they are each large problems requiring large investments of

their own. More importantly, virtual manufacturing is already the subject of several research and development efforts, and there are numerous new commercial products in the PDM area, as well as substantial in-house PDM efforts by the major airframers. The issue of leverage is very important. Our focus in the study has been to identify the technologies that will yield high paybacks on modest research investments. Improvements derived from advances in PDM and virtual manufacturing are going to happen—the important questions that we have addressed are: What are the other technologies that should be employed? How can we bring other technologies to bear on the Automating Maintenance Instructions (AMI) authoring problem? and, How can these new AMI capabilities be integrated into the new PDM/CAD/CAE system design environments? The answers to the first and second questions are the subject of this report.

A short answer to the question of the integration of new AMI technology-based capabilities into the PDM/CAD/CAE system design environment can be provided here. Our expectation is that the architectures for the new system design environments will at least support client-server operations and, in the future, can be expected to move toward a distributed object architecture. The new AMI capabilities, based on the technologies discussed in the study, can be configured to operate from servers in a client-server environment or as objects in a distributed object environment. They will form resources or components providing services to newly developed PDM/CAD/CAE AMI authoring environments.

This study of technologies for AMI closely follows two recent studies of the authoring of T.O.s: Zimmerman et al. (1993) and Sanchez et al. (1997). The Zimmerman study examined a broad range of technologies that can be expected to support the creation and management of maintenance instructions. The technologies examined included product data and product data management, qualitative physics, automated planning, human modeling, automated text generation, graphics synthesis, automated verification and virtual reality. Recommendations addressed the short range, medium range and long range potentials of the technologies examined with a focus on the development of a research agenda to support the medium range agenda. The Sanchez et al. (1997) study addressed more closely the role of PDM, including design and manufacturing simulation tools, in the authoring of Interactive Electronic Technical Manuals (IETM). It focused on procedural, as well as testing and troubleshooting tasks.

The goal for our Automating Maintenance Instructions study has been to identify technologies that can be assembled to contribute to the automated generation of technical data for maintenance instructions. To launch the study effort, two technology assessment sessions were organized, the first at the University of Pennsylvania and the second at BBN Technologies. The objectives of these sessions were to collaboratively develop a vision for AMI, catalog the technologies that might be further developed or employed to realize the AMI vision, and lastly, determine the subset of technologies from which the Air Force might expect its greatest return on investment.

With the set of AMI technologies identified, the study was then organized around the T.O. author's view of a hypothetical AMI system. Given that the author has selected a maintenance procedure to develop, how can the technology-based automation most

effectively support the author in the development and validation of that procedure? Broadly stated, our goal was first to automatically generate and propose a maintenance procedure that meets the author's objective as closely as possible, and then provide the author with the graphical and textual material as necessary to edit, complete and validate the procedure. Given these objectives, we then broke out the problem into two case studies. In Case Study 1, BBN Technologies addressed the problem of generating the proposal of a procedure to meet the author's requirements. In Case Study 2, the University of Pennsylvania addressed the problem of generating the textual and graphical material to be made available to the AMI author to support the revision and validation of a maintenance procedure.

Case Study 1, addressed by BBN Technology, forms the input side of the AMI technology study. The question asked is: How can technology be used to best provide input to the authors of T.O.s in building maintenance procedures? The author's goal is to produce a complete and validated procedure. Hence, the question becomes: Can technologies be assembled to automatically plan, construct and propose a procedure that satisfies the author's stringent requirements? The plan must describe the procedure to executed at the appropriate level of detail and, with perhaps minor editing by the author, and then be ready for the validation process. It would be highly optimistic to expect that the proposed procedure would be fully satisfactory, but it would have to be close to satisfactory most of the time if it is to be a significant productivity aid. This is a planning problem. CAD, and to a lesser extent, CAE sources are the principal input data sources to be considered. They will provide important input to the planning process, but as important as their input is, there are significant gaps in their ability to fully meet the needs of the maintenance procedure planning process. CAD and CAE are not of much help in determining that a cavity enclosing fuel system components should be purged before working within the cavity, they do not help much in determining that three rather than two maintenance personnel are required to safely execute a procedure, and they do not offer much help in determining that a bucket should be used to catch residual fuel as a valve cap is removed. They do not offer much help in selecting particular cautions, warnings, and notes that insure the safety of personnel, prevent damage to expensive equipment or assist in the efficient execution of the required procedures. In today's T.O. development environment this is critical information that the author develops using not CAD data, but his or her domain knowledge, extensive experience, and common sense. Building these skills into a planner is a formidable problem.

The significant shortfalls in the ability of PDM/CAD/CAE to provide all the information necessary to the authoring process are critical. If we are to recommend the use of a planner as suggested in Zimmerman et al. (1993), this shortfall must be addressed. The development of a generative planner holds considerable promise, but requires an extensive front end knowledge acquisition effort. This does not preclude the use of a generative planner and we do, in fact, suggest that a generative planner be a component in the planning capability for AMI. In finding a way to complement the capabilities of a generative planner, two observations are important. The first is that for most large, new Air Force systems and more particularly, new aircraft, there is considerable reuse of system and subsystem components—from a maintenance perspective there is much that is

not new. The second observation is that existing T.O.s, as the product of many hundreds of person-years of effort are a potential source of the information not available in the PDM/CAD/CAE data. These two observations, taken together, suggest that existing T.O.s are a suitable target for a data mining operation as a means to develop a planner for AMI. To move in this direction, we have explored the use of linguistic analysis and natural language understanding to obtain computer-based procedure representations from existing T.O.s. The procedures derived in this manner would then be indexed for use by a case-based planner. A hybrid planner, with case-based and generative components, is suggested as the means to build maintenance procedures in response to requests from the AMI author.

In developing Case Study 2, the University of Pennsylvania has reexamined its extensive work in natural language text generation and human figure modeling from the perspective of the goals for AMI. They have examined the linguistic forms taken by the diverse portions of a T.O.: the procedure steps, cautions, warnings and notes. Their linguistic analysis of a significant F-16 maintenance procedure corpus has been used to support their analysis of text generation requirements of the AMI author and has also been used to support the BBN language understanding effort. Their research conducted within the study confirms their assertion that their text generation capabilities can produce the textual material to support the AMI author in generating T.O. procedures.

The University of Pennsylvania study also addressed the assertion that the animation of maintenance procedures using a human figure model has the potential to support the AMI author in the development and validation of maintenance procedures and can also be used as an important part of the T.O.s themselves. However, the level of description in the T.O.s targets the maintenance person who brings considerable real-world skills to the task at hand. In contrast to the skilled maintenance person, the human figure model must be told even the intuitively obvious steps of where to stand and which way to look to reach a given part. In this study, the University of Pennsylvania provides insight into the research underway to bridge this gap, and thereby provide the human figure model with the capability to execute maintenance procedures based on a computer representation of their textual description. We speak of Case Study 2 as the output side of AMI. It addresses the development of textual and graphical data to be made available to the AMI author.

AMI Technology Assessment

At the outset of the study it was decided that it would be useful to collaboratively generate a vision of what is feasible to accomplish in the domain of AMI within a 5 to 10 year time frame. Accordingly, two workshops were convened, one at The University of Pennsylvania and one at BBN Technologies. At both workshops there were representatives of computer-based plan representation, agent technology, human factors issues, linguistic analysis, language understanding and automated text generation. In addition, at the University of Pennsylvania workshop there were also representatives of human figure modeling technology. At the BBN workshop there were also representatives of intelligent tutoring and case-based reasoning technology.

The following list enumerates the elements of the vision produced initially through discussions at the University of Pennsylvania and subsequently augmented by discussions at BBN.

1. **Product Model Drives Technical Order Generation.** It is expected that product design will be accomplished within a computer-aided design system. It was argued that the physical specifications represented in the CAD system should drive not only the design itself, but also the development of the Technical Orders supporting the system.
2. **Technical Orders in the Future will be Derived from an Integrative 3-dimensional, Multimedia Computer-based Representation.** The workshops envisioned a unitary procedure representation that starts with the product model, incorporates tasks, procedures and constraints associated with disassembly, assembly, removal, testing and repair and includes rules for instruction presentation, cautions and warnings, and notes.
3. **Model Scope Expanded via Text Analysis of Existing Technical Orders.** There will be many features of a T.O. that will not be a part of any standard CAD representation, even in "smart" CAD systems of the future. Accordingly the vision suggests augmenting the product model by undertaking text analysis of existing, previously prepared T.O.s in order to detect needed features of new T.O.s. These text analyses will be most useful if the existing orders being analyzed are as close as possible in domain content to the target order. Text analysis only needs to be done once per domain to identify needed features.
4. **Deliver T.O.s via VRML Web Pages on Hand-held, Wireless, Speech Controlled PDAs.** Universal accessibility, including 3-D imagery, will be supported by VRML web pages. The usability to the maintenance technician is enhanced by using hand-held wireless PDAs using speech as an input mode.

The second workshop task produced a list of specific technical developments and process changes necessary to support the vision.

1. **Computer-based Procedure Representation.** Developing a general methodology with which to represent maintenance procedures in terms of abstractions that are compatible with CAD representations on the one hand and have the potential to be converted to graphics and natural language on the other will be a significant challenge.
2. **Natural Language Generation.** Natural language generation capabilities will be needed to generate the text messages that will be required in T.O.s. Language generation refers to the process of producing natural sounding text or speech messages from abstract coded information derived from a number of sources.
3. **Author Interface.** It was anticipated that even after ten years fully automated T.O.s would still not be feasible. In fact, it was acknowledged that this was an unrealizable goal. Accordingly there is a need to generate an AMI interface through which the author will interact with the procedure representation.

-
4. **Multimodality Instruction Presentation.** It is anticipated that the final T.O.s will be presented in rich multimedia formats for ease of interpretation and understanding
 5. **Maintenance Process Planner.** Automatically generated T.O.s will require the introduction of automated planning technology to assist in determining the priority and order in which the various task elements must be accomplished. Both **case-based planning** and **generative planning** were considered.
 6. **Labeled CAD World.** In order to be useful, CAD product representations must be formed from semantically labeled objects that are decomposed into their constituent sub-objects that are meaningful from the perspective of assembly and repair.
 7. **Natural Language Understanding.** Language understanding technology is required in order to make use of existing T.O.s as a source of information for articulating future T.O.s.
 8. **Procedures Description in Language.** Once encoded in the computer, it is a further step to automatically convert maintenance procedures from abstract code into terms that can be expressed in language and flow charts. This step is necessary before a natural language generation program can actually produce fluent text.
 9. **Specification of Assembly/Disassembly.** It is not obvious from the data in a CAD diagram how parts are disassembled or assembled. There are constraints on what must be removed to get access to other parts, sequential constraints, and procedural constraints.
 10. **Specification of Physical Constraints.** In addition to the assembly/disassembly constraints there are alignment constraints, pressure constraints, torque constraints, etc. that must be specified.
 11. **Specification of Safety Constraints.** In addition to the above constraints, repair comes with cautions and warnings about safe ways to accomplish specific tasks and prevent damage to equipment.
 12. **Knowledge Acquisition for labeling.** The labeling requirement involves extensive knowledge acquisition to determine and capture existing naming conventions. It would be useful to have technology to support collection of this information.
 13. **Task-level Physical Agent Models.** The T.O.s will need to show demonstrations, either static or animated, of maintainers performing selected tasks. Models are needed that allow the execution of task level performance by animated mannequins.
 14. **Measures of Task Characteristics (e.g., Complexity, Time-to-Complete).** Guidance will be needed to select among alternative ways of accomplishing maintenance tasks. Performance measures will be needed that will provide the indices against which to evaluate and select alternative methods.
 15. **Object-oriented Action Description.** Maintenance tasks will be broken down into actions. Actions need to be segmented and treated as programming objects so that they have attributes associated with them such as, which hand they should be

performed with, where the eyes should be directed, and whether this is a bench or field operation.

16. **Simulation of How Things Work.** In addition to assembly/disassembly operations, T.O.s will contain information about how components and assemblies of components work. To reason about these activities requires the ability to simulate their performance.

After brief introduction and discussion of these technology innovations, each group was asked to rate each innovation, using a Delphi procedure, with respect to the following dimensions. A total of 10 individuals contributed ratings to the analyses presented below.

1. **Importance in Contributing to the Process.** How important is this innovation to achieving successful automated technical order generation?

Scale: 1 to 5 where 1 is extremely important and 5 is not important at all.

2. **Importance of Air Force Investment.** How important is it that the Air Force fund development of this technology for the desired result of demonstrated automation of technical orders to be achieved?

Scale: 1 to 5 where 1 is extremely important and 5 is not important at all.

3. **Risk that Applying Resources to this Innovation will Result in Useful Outcomes**

Scale: 1 to 5 where 1 is very risky and 5 is not risky at all.

4. **Time Scale Over Which to Expect Results.** If a push to develop the required implementation of the technology were to start today, how long would it take to achieve meaningful results in terms of implementing automated T.O.s

Scale: 1 year to 20 years.

According to the Delphi procedure each individual first rated each innovation on each dimension. Then a discussion of the rating was held during which large disagreements were highlighted and each rater had an opportunity to say why he or she had so rated the innovation. Then each rater was given the opportunity to revise the rating, based on the discussion.

The revised ratings that resulted from this process are summarized in the following tables.

Table 1. Technology Innovations Ranked by Average Importance
(1.00 is Extremely Important; 5.0 is Unimportant)

	Average Importance
Computer-based Procedure Representation	1.00
Natural Language Generation	1.00
Author Interface	1.00
Multimodality instruction presentation	1.17
Case-based Planner	1.22
Label CAE World	1.36
Domain-specific language understanding	1.45

Generative Planner	1.56
Process description in language	1.56
Specification of assembly/disassembly	1.64
Specification of physical constraints	1.73
Knowledge Acquisition for labeling	1.73
Task-level physical agent models	1.73
Specification of safety constraints	1.91
Measures of task characteristics	2.09
Object-oriented action description	2.09
Simulation of how things work	2.55

Table 2. Technology Innovations Ranked by Average Importance of Air Force Investment (1.00 = Extremely Important; 5.0 = Unimportant)

	Average Import. for AF
Computer-based Procedure Representation	1.50
Natural Language Generation	1.50
Multimodality instruction presentation	1.60
Case-based Planner	1.67
Domain-specific language understanding	1.73
Author Interface	1.80
Process description in language	1.89
Generative Planner	2.00
Specification of safety constraints	2.10
Task-level physical agent models	2.10
Object-oriented action description	2.10
Knowledge Acquisition for labeling	2.20
Measures of task characteristics	2.20
Simulation of how things work	2.64
Specification of physical constraints	2.70
Label CAE World	2.80
Specification of assembly/disassembly	2.82

Table 3. Technology Innovations Ranked by Average Risk
(1 = very risky; 5 = not risky)

	Average Risk
Specification of assembly/disassembly	3.73
Knowledge Acquisition for labeling	3.73
Computer-based Procedure Representation	3.64
Author Interface	3.50
Natural Language Generation	3.45
Label CAE World	3.40
Case-based Planner	3.38
Domain-specific language understanding	3.27
Object-oriented action description	3.20

Process description in language	3.11
Measures of task characteristics	3.05
Specification of physical constraints	3.00
Simulation of how things work	2.91
Task-level physical agent models	2.90
Multimodality instruction presentation	2.83
Specification of safety constraints	2.55
Generative Planner	2.38

Table 4. Technology Innovations Ranked by Average Estimated Time (Years) to Achieve Results

	Average Time Scale in Years
Specification of physical constraints	2.50
Computer-based Procedure Representation	2.50
Specification of safety constraints	2.70
Multimodality instruction presentation	2.83
Knowledge Acquisition for labeling	2.90
Measures of task characteristics	3.00
Author Interface	3.00
Domain-specific language understanding	3.05
Object-oriented action description	3.20
Natural Language Generation	3.23
Case-based Planner	3.38
Specification of assembly/disassembly	3.50
Task-level physical agent models	3.89
Process description in language	4.11
Label CAE World	5.30
Generative Planner	5.88
Simulation of how things work	6.60

Before discussing the individual ratings it is informative to examine the correlations among various average rating components to determine the extent to which the judgments were independent. Each correlation is calculated over the average ratings of sixteen technical developments for each pair of scales. These correlations are shown in Table 5. Since the correlation matrix is symmetric, only the six unique correlations are shown. As shown in Table 5 there is a relatively high correlation between *importance* and *impact for the Air Force*, moderate correlations between *the time scale* judgments and the other dimensions and a moderate correlation between *importance* and *risk*. There was essentially no correlation between *impact for the Air Force* and *risk*. The negative correlations result from the fact that the risk scale is inverted in comparison with the others. We conclude that the raters had difficulty separating out *importance* from *Air Force impact*, but the other judgments were relatively independent.

Table 5. Correlation Among Technology Innovation Ratings

	Average Impact for AF	Average Risk	Average Time Scale
Importance	0.60	-0.39	0.37
Impact for AF		-0.02	0.40
Risk			-0.35

The innovation judged most important in general and for Air Force impact was the need to develop *computer-based procedure generation* technology. Since maintenance instructions are basically procedures, it is not difficult to understand why this was considered so important. It is worth noting that this innovation was also rated as high-risk. Also receiving high rankings was *natural language generation* capabilities and *multimedia instruction* generation. These alternatives both have to do with translating a computer representation into usable output, also an obviously important aspect of the problem. *Case-based planning* was ranked highly in both domains as well. We believe this ranking is because the group felt that this was a promising approach that could make a significant contribution. *Generative planning* was ranked lower down the list. It is interesting that *Labeling the CAE World* was rated highly on importance, but very low on Air Force Impact. We believe that is because the raters were predicting that this was a development that would happen whether the Air Force supported it or not.

Risk ratings reflect the certainty with which the raters thought the respective innovations were achievable. Besides procedure generation, high risk technologies were *specification of assembly/disassembly*, *knowledge acquisition for labeling* and the development of an *author interface*. It would seem that creation of an author interface would be straight forward. We believe the reason this was rated high risk was because it is not clear at this point exactly what tasks would be assigned to the author and how this person will be expected to interact with the automated system.

Finally, with respect to development time scale, the innovations places at the most extreme end of the scale (6.6 years) were *labeling the CAE world*, producing a *generative planner* and creating *simulations of how things work*. In the case of the CAE labeling, this is probably because it was forecast that industry will take on this task itself and that they will not be driven by urgency. It is difficult to interpret the long lead time for a generative planner when this was labeled low risk. Perhaps it was interpreted that high priority will not be given to it and therefore it will take a long time. It is encouraging that computer-based procedure representation was scored as being achieved relatively quickly (2.5 years).

In summary, we have defined the elements of a vision for automated Technical Order generation and assessed the priorities for technological innovations that will be required to achieve that vision. This analysis served as a basis for the remainder of the study.

2. AMI Input Side Capabilities and Supporting Technologies

The Technical Order author's principal task is the construction of a tested and verified maintenance procedure. Given the extensive work on PDM and the increasing integration

of CAD and CAE capabilities by commercial vendors, the airframers and their vendors, we must assume that the authoring of Technical Orders will become part of this environment. The most significant benefit to the author and one that addresses what has been a fundamental and costly problem area will be ready access to CAD and CAE data.

In contrast to the soon-to-be-solved mechanical problem of providing access to CAD and CAE data, the construction of a complete and accurate Technical Order presents significant human factors and engineering challenges. Any steps taken to automate this construction process can be expected to assist the author in moving more quickly to a completed procedure and yield significant cost savings to the Air Force. Case Study 1 focused on assembling the capability to automatically construct procedures that can then be provided to the author for completion and validation. In developing a procedure for a Technical Order, the author would have available a proposed procedure for the task and an authoring environment within the PDM/CAD/CAE environment in which edit and validate the procedure. The ability to propose procedures for the author is based on a broad range of technologies. The technologies include computational linguistics, plan representation and automated planning. The automated planner must draw on data from the CAD and CAE data bases. But much of the material that goes into a procedure is derived from the broad range of experience and common sense knowledge of the authors of Technical Orders. The resources that hold this invaluable "data" are existing Technical Orders. This important input to procedure construction can be obtained for the planner via data-mining from existing Technical Orders.

Figure 1 provides an overview of the process and technologies required to support the automated construction of a proposed maintenance procedure. CAD and CAE and existing Technical Orders are the principal inputs to the process. Linguistic analysis and natural language understanding are used to construct the initial plan library from existing Technical Orders and a planner creates a proposed procedure at the request of the author. Finally, new procedures are indexed in the plan library for future use. Each of these processes and technologies uses and operates on a common representation of the procedure under construction. The contribution of each of these technologies is discussed in the following sections.

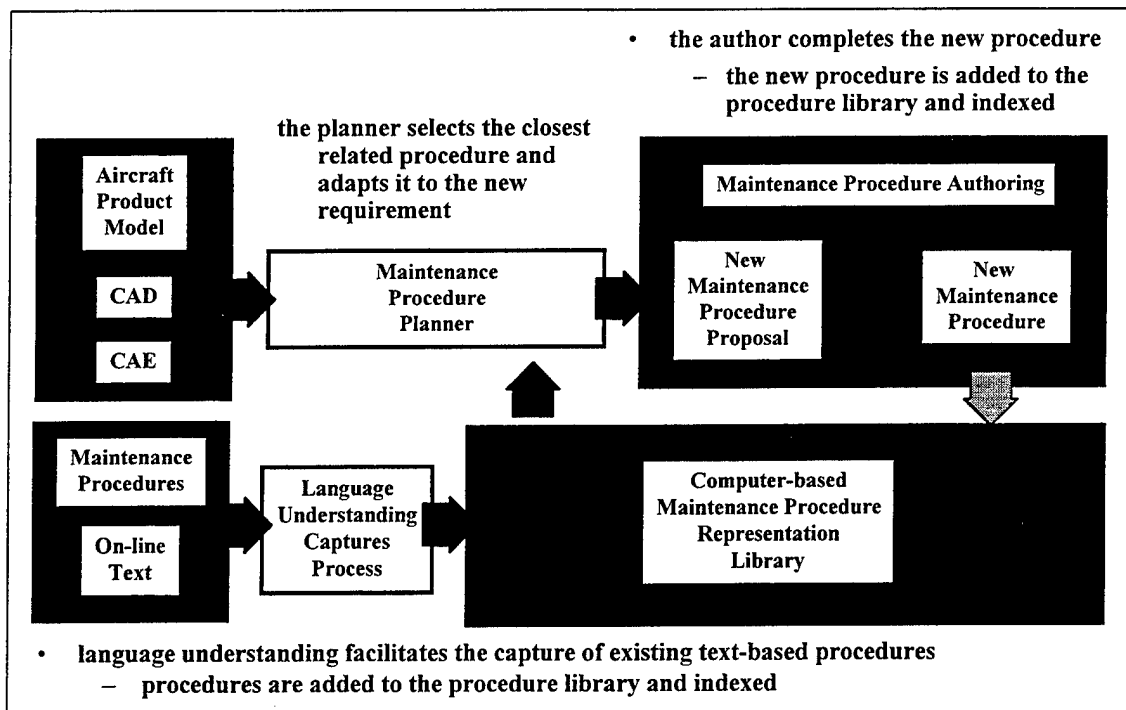


Figure 1. Automation for Maintenance Procedure Authoring

2.1 Technical Order Procedure Representation

Today, most T.O.s fill the pages of loose-leaf binders. They are impressive in their quantity—a subset of the C-Shop manuals for the F-16 fills all of a large filing cabinet. The principal users are the maintenance personnel for the documented system. For the F-22, T.O.s are available on a Portable Maintenance Aid (PMA), an 18 pound laptop computer. The form of the F-22 T.O.s closely follows the form of the paper F-16 T.O.s. In each of these cases, the authors of the T.O.s are preparing, essentially, paper-based documents for maintenance personnel. The plan representation for a T.O. is typically English language text with exploded views of relevant parts. The text and exploded views are presented on facing pages of the manual. The procedures of a T.O. are designed for human consumption—they do not have a computer-based representation.

The AMI technologies require a basic change in T.O. plan representation. The maintenance personnel remain the principal users of the T.O.s and it is still the authors that have final responsibility for creating them, but the plans must also be available in a form that can be operated on by software programs, the technologies for AMI. The plans must take a form that can be operated on and used by both people, maintenance personnel and authors, and computer programs. Authors need to be able to build and edit a T.O., while maintenance personnel need access to the final product of the authoring process. The authors' primary concerns are the content and form of the T.O.s that they are developing. As they develop procedures, authors need the capability to view the final form that the T.O.s will take. The procedure representation must be an adequate base from which to generate the T.O. format.

The ability to generate a T.O. in final form based on the procedure representation is the final step in the AMI process. The broad range of technologies (see Figure 2) that has been recommended to support the authors of T.O.s will each depend on the procedure representation. The process starts with linguistic analysis and language understanding. The linguistic analysis will identify the actions and objects of the domain that the procedure representations must address. The language understanding through the use of a parser will provide the capability to translate the English language descriptions of relevant procedures into a computer-based procedure representation using the lexicon of actions and objects previously identified in the linguistic analysis. The indexer in the case-based planner, using information provided by the parser, will operate on the procedure representation to make it a part of its case-base. At plan proposal time, a procedure will be retrieved, based on the indexing, and adapted to meet the author's requirements. Using an interactive authoring environment, the author will review the proposed plan and make any necessary changes by operating on the procedure in final form, while actually updating the underlying procedure representation. The last step is to index the new plan in the case-base for future use. Each of these steps in the process of supporting the authoring of T.O.s is closely linked to the procedure representation. Taken together, they emphasize the importance of the role of procedure representation in the authoring process and point to the central role for plan representation in AMI system design.

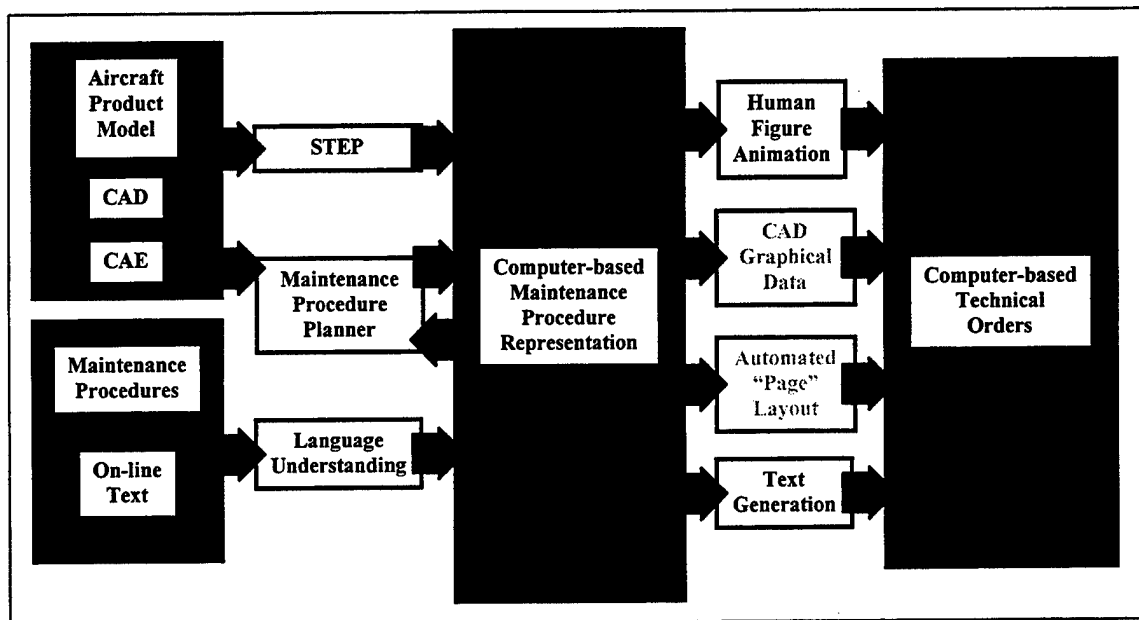


Figure 2. Procedure Representation as a Central AMI Capability

2.1.1 Maintenance Procedure Representation

Computer-based maintenance procedure representation must capture the detailed content of T.O.s. The emphasis will focus on the linguistic content of the procedure. The references to the physical objects of the procedure and actions taken on those objects are expected to provide the basis for selecting the graphical content necessary to complete the

procedure presentation. The principal concerns are what operations are performed on what entities and the order in which these operations should be executed. Many procedures require more than one person, hence the allocation of steps among the personnel is important. The basic instructions of the procedure are supplemented by cautions and warnings to insure the safety of personnel and prevent damage to equipment. Notes provide additional supplemental information.

Procedure representation falls within the broader field of knowledge representation, an active research area for many years. Hence, there are several candidates that might form the basis for procedure representation for AMI. There are a number of correct choices that might be made among those available. For the purposes of the study, two representation frameworks were utilized, one in the research conducted by BBN Technologies and one in that conducted by the University of Pennsylvania. The Operator Model Architecture (OMAR) was selected by BBN based on its successful experience in using the OMAR representation languages for closely related work (Deutsch, MacMillan, Cramer, & Chopra, 1997). In the University of Pennsylvania case, the Parameterized Action Representation (PAR) is being developed specifically to address the applications of natural language generation and the animation of maintenance procedures using a human figure model.

Within the framework of the study it is useful to look at more than one alternative as the basis for procedure representation. It has also been made necessary by the need to provide demonstrations of the key technologies. The University of Pennsylvania's selection of the PAR was based on its capability to support both its text generation and human figure animation portions of the demonstration, while BBN's selection of OMAR was based on the ability to interface it to its natural language understanding system, case-based planner, and simulator. Ideally, in building an AMI system a single procedure representation should be sought. However, as discussed below, this is not an essential requirement for progress in AMI.

2.1.2 OMAR Procedure Representation

The Operator Model Architecture (OMAR) (Deutsch, Adams, Abrett, Cramer, & Feehrer, 1993; Freeman, 1997) was designed as a test bed for human performance modeling (Deutsch & Adams, 1995). Many of its components have an important role to play in supporting T.O. automation. As a human performance modeling framework, OMAR is used not only to model the human players in a simulation environment, but also the entities that they interact with. It has frequently been used to model problems in the civilian air traffic control environment where the models have included air traffic controllers at their radar-based workstations and flight crews on their aircraft flight decks. Problem exploration has involved examining the execution of flight crew and air traffic controller procedures. OMAR provides a simulator for the execution of these procedures and analysis tools that enable the evaluation of simulation runs. It is this ability to represent procedures and simulate the execution of procedures that is of primary importance. The AMI author is concerned with developing maintenance procedures—

OMAR provides one approach to managing the computer-based representation of those procedures as they are developed and edited in the authoring process.

OMAR has two languages that form the basis for representing human behaviors—in the AMI environment, the procedures for carrying out maintenance procedures as set out in T.O.s. A frame language, the Simple Frame Language (SFL), is used to represent and describe the entities or objects in the environment. The major elements to be represented will be aircraft parts and the tools used in carrying out maintenance procedures. Much of the descriptive portion of the representation can be derived from CAD data. The information that may be represented can include part *type* data as well as information on particular parts. AMI must also be concerned with operational information with respect to parts. Some parts are disposable, most are not. It is appropriate to purge a vent tank, but few other things are purgeable. Some of this operational information can be determined from CAD data, much of it can be determined as part of the language understanding process applied to existing T.O.s. Additional requirements on the representation language include the ability to represent *part-of* and *has-parts* relationships and the connectivity of parts. SFL easily meets these requirements. SFL's multiple inheritance capability makes it possible to establish part type hierarchies and operational characteristics simultaneously. *Concepts* define the entities described using SFL. The concepts have *slots* that define the attribute-value pairs particular to the entity.

Procedure representation must also be concerned with who does each step in a procedure. While some procedures can be carried out by a single maintenance person, many require the coordinated activities of several maintenance personnel. The procedure representation must also include a representation of the agents, the maintenance personnel, who carry out the procedure. Within OMAR, the agents are defined as SFL entities. They can be assigned the skill levels appropriate to the maintenance tasks being undertaken.

It is readily apparent that the number of entities that a T.O. author might have to deal with for an aircraft will run into the thousands. While some of the capture of this data for an AMI system can potentially be automated, it will be necessary for the AMI system developers to have software tools available to oversee and manage this process. OMAR provides a graphical editor for SFL that enables both individual concepts and the network of concept definitions to be reviewed and edited. The SFL graphical editor is a tool for the system developer (not than the T.O. author). SFL and its graphical editor are representative of the entity definition frameworks that can be used in the development of an AMI system.

SFL addresses entity definition, but not process or procedure definition. The second OMAR language, the simulation core (SCORE) language, is used to define procedures. SCORE procedures are themselves SFL-defined entities making it easy to categorize procedures along several dimensions. We can categorize them by the systems they address, for instance the fuel system. Further specializations can be used to make it possible to associate particular caution or warning messages with particular classes of procedures.

For the most part, the individual steps of a procedure as executed by a maintenance person are sequential, however some procedures have steps which may specify coordinated actions as in adjusting a valve to establish a particular pressure in a system. Other procedures require the coordinated actions of more than one maintenance person—one person might trigger an event while another observes the response of an indicator in a flight deck instrument. There are forms in the SCORE language making it possible to represent these relationships among the personnel executing a procedure. A given maintenance procedure will have one or more agents corresponding to the number of personnel that the procedure requires. A SCORE procedure would describe the primitive steps of a procedure such as *remove* or *disconnect*. Arguments to the procedure would specify the entities operated on.

The default behavior for SCORE procedures, to execute subprocedures sequentially, addresses the typical execution pattern for most procedures. The SCORE language also includes *race* and *join* forms to define the parallel activities of a single person or the coordinated activities of several personnel. Within a *join* form, all the activities execute to completion, while within a *race* form, all the activities are terminated as soon as the first one has completed. A *join* form can be used to specify that two personnel must complete their current steps before any subsequent steps are taken, while a *race* form can be used in adjusting a valve until a specified pressure is established—when the pressure is observed to reach the specified pressure that branch of the race completes and under the terms of the race form, the adjusting of the valve is terminated. In contrast to the sequential steps of many maintenance procedures, diagnostic procedures do extensive condition testing and branching. As in most computer languages, SCORE provides a set of condition testing forms.

The SCORE language component within OMAR provides the capability to represent the broad range of maintenance procedures that an AMI system must address. The SCORE language is not the level at which the AMI author should work, but the language provides the representation capability required by an AMI system. To aid the AMI system developer, a graphical browser is available that provides both a view of individual procedures and the calling patterns of a network of procedures. Figure 4 provides a procedure browser view of the initial steps in F-16 Procedure 2-14-1 (see Figure 3), the Removal of the Internal Fuel Tank Vent and Pressurization Valve. Each of the subprocedure calls corresponds to a step in the procedure. Some of the steps are atomic actions, such as, the individual *remove* and *purge* steps, while others are composites of two or more operations, as in the *remove-and-slide* step. In capturing existing procedures, it will be important to capture the level at which they were expressed in the T.O. of the procedure, that is, the level at which they are shown in Figure 4 in this example.

2-14-1. REMOVAL OF INTERNAL FUEL VENT AND PRESSURIZATION VALVE

NOTE

All serviceable parts will be retained for reinstallation.

1. (A) Remove access panel 3405. (General Maintenance)
2. (A) Purge vent tank. (T.O. 1-1-3)

NOTE

Protective devices shall be installed on all open tubes, ports, and electrical disconnects.

3. (A) Remove coupling and slide sleeve on elbow.
4. (A) Rotate elbow to provide clearance around valve.
5. (A) Disconnect and remove pressure sense tube.
6. (A) Remove coupling and slide sleeve on pressurization tube.

-CONTINUED-

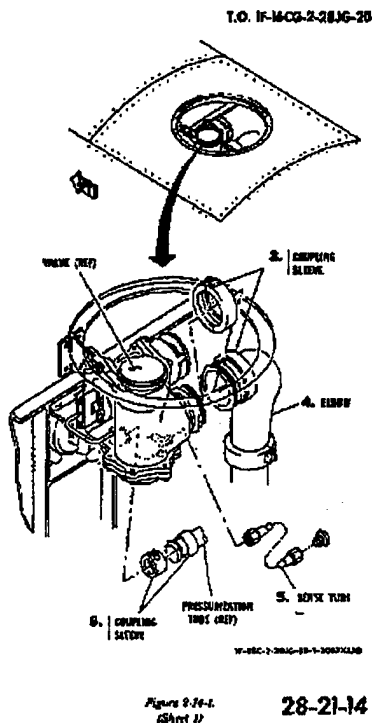


Figure 3. F-16 Fuel System Procedure 2-14-1 (Page 1)

In an AMI system, the SCORE representation of the procedures that form the case base would be generated by the parser. Sparser, the parser being used for the Case Study 1 Technology Demonstration, provides the capability to add the code necessary to generate the SFL and SCORE forms that, in turn, generate the entity and procedure definitions that are the output of the parsing process. In the interactive editing environment that the AMI author would use to revise procedures, it is the SCORE procedure representation that would change as a result of the editing process. The form that the procedure would take while being edited by the AMI author would be derived from the SCORE representation. The author would not be working with, nor ever be aware of, the SCORE representation itself.

2.1.3 Simulating Procedure Execution

Simulation can play a broad range of important roles in supporting the AMI maintenance procedure author. In Case Study 2, the University of Pennsylvania has examined the use of a human figure model to simulate the execution of a procedure. In Case Study 1, our goals were more limited. We have suggested that language understanding can be used as a data mining process to capture existing maintenance procedures that can then be adapted for use by the AMI author in developing new maintenance procedures. The continuity between language understanding and case-based planning is provided by the plan representation, a representation that can be shared by these two system components. The plan representation can also be used to support the simulation of the procedure. By using the OMAR SFL and SCORE languages for plan representation we gain access to

the OMAR simulator that enables the procedures to be played out—the same plan representation directly supports procedure simulation.

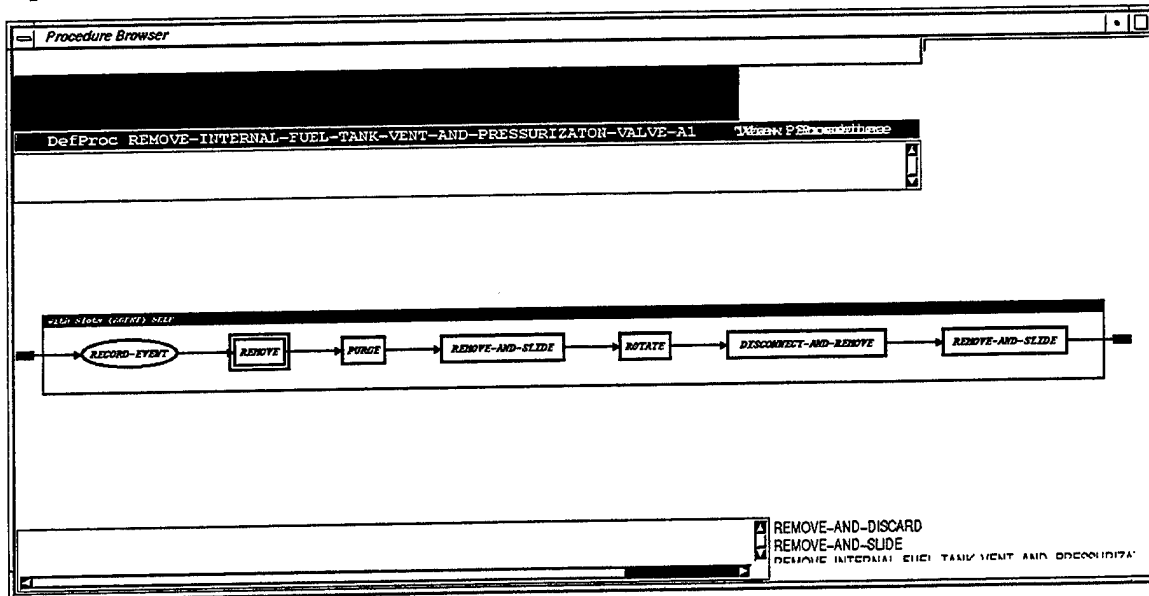


Figure 4. Procedure Browser View

In Figure 4 we saw a Procedure Browser view of one of the F-16 fuel system procedures. When played out in the simulator, the execution of the procedure generates the trace that is shown in Figure 5. As noted earlier, the procedure is carried out by two maintenance personnel. In the simulation, as shown in the trace, they are labeled MAINTENANCE-AGENT-A and MAINTENANCE-AGENT-B corresponding to the (A) and (B) labels that appear in the T.O. (Figure 5) denoting the particular person executing each procedure step. As shown, the procedure is executed primarily by MAINTENANCE-AGENT-A. MAINTENANCE-AGENT-B steps in to disconnect the electrical connector P4 at time 130 and then assists MAINTENANCE-AGENT-A in removing the bolt and washer from the mounting bracket at time 150. The remaining steps of the procedure are completed by MAINTENANCE-AGENT-A. For the purposes of the Case Study, no attempt was made to assure that the times assigned to individual steps were realistic—they do however, show the coordination of the actions of the two maintenance personnel involved in the execution of the procedure. For the purposes of the Case Study, trace statements were included at the level of the individual actions of each maintenance person. In the T.O. for Procedure 2-14-1, step three is presented as “Remove coupling and slide sleeve on elbow.” In the trace, the two actions of removing the coupling and sliding the sleeve appear in separate trace lines. It will be important to correctly capture this level of detail that is readily available from the parser.

The OMAR simulator also includes a number of analysis tools used to evaluate simulation runs. One of these analysis tools is a time-line display that provides a Gantt chart style representation of the execution of the procedures by each simulation agent. Figure 6 shows the time-lines for MAINTENANCE-AGENT-A and MAINTENANCE-AGENT-B in executing F-16 T.O. 2-14-1 (Figure 3). The Gantt chart output shows the

respective timings of the actions of the two personnel coordinating their activities in the execution of the procedure. In contrast to the on-line trace (Figure 5), this view of the procedure execution retains the original form of step three as a composite of removing the coupling and sliding the sleeve, that is then broken out into its component steps. The representation of the procedure includes both levels of representation: the original form stated as a composite of removing the coupling and sliding the sleeve as shown only in the timeline, and the breakout into individual steps as shown in both the timeline and the trace.

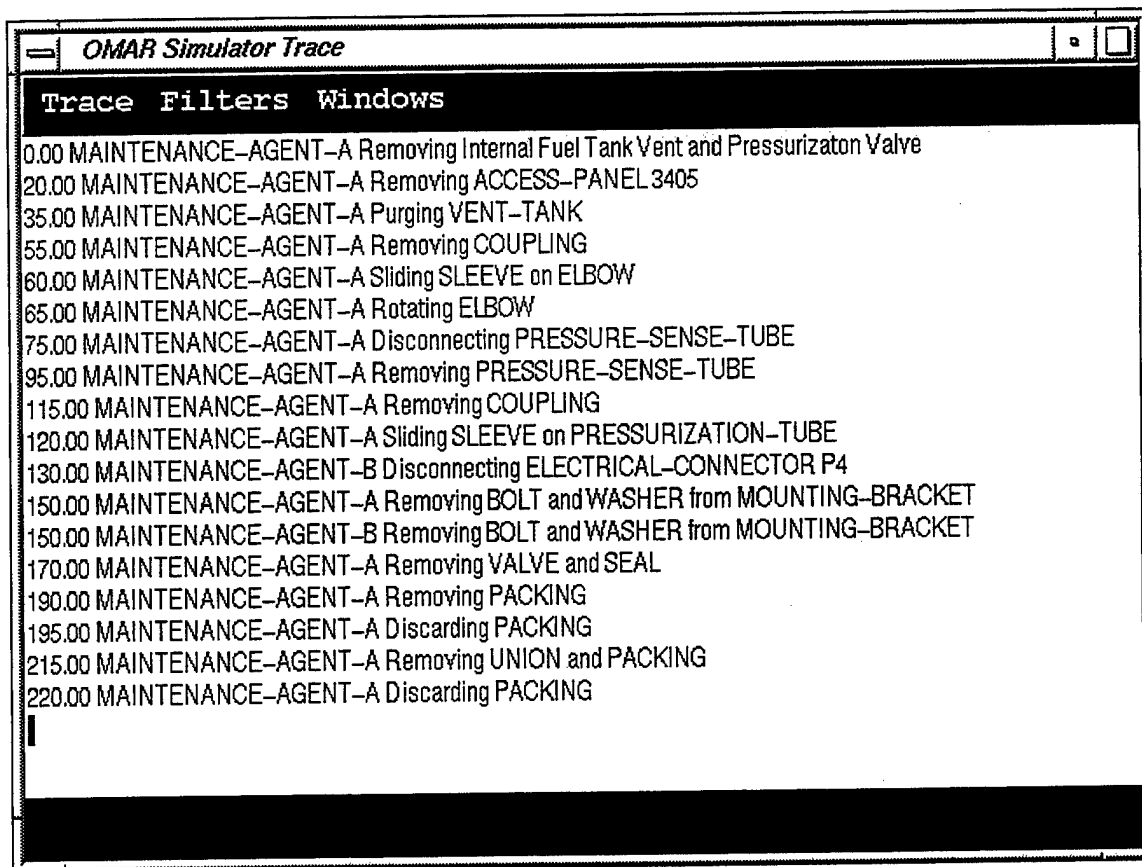


Figure 5. Maintenance Procedure Trace

2.1.4 One Procedure Representation or Several

In Case Study 1, the OMAR procedure representation is suggested as a vehicle for integrating the several technologies that are advocated for the input side of AMI. In Case Study 2, the Parameterized Action Representation plays a similar role for the Output Side of AMI. This clearly raises the important question of just how many procedure representation languages there should be. It also brings into focus the more basic question of whether a representation language can actually fulfill the integration role suggested for it. The classic problem underlying this issue is that of integrating legacy systems to achieve new objectives. In some cases, the legacy systems can readily be adapted to share a given representation, while in others the sharing is not easily accomplished.

For the input side of AMI we believe that the OMAR representation can perform the integrative function of providing a representation that can be utilized by the several input side technologies: language understanding, knowledge representation for case-based and generative planning, and simulation. For the output side of AMI, the Parameterized Action Representation is being developed to support both the human figure simulation of maintenance procedures and the generation of the textual descriptions of the procedures.

In seeking to resolve this issue, it is important to keep our goal in mind: to provide an integrated suite of technologies that can play together in an AMI authoring environment that fits within a large scale CAD-based system design environment. This goal dictates that the technologies of the input and output sides of the AMI system play together—their respective technologies require access to procedure representation. Do they all have to share the same representation? Probably not. It would be nice, but it is not necessary. The input side technologies share a single representation, while on the output side, in addition to the PAR, the Sentence Planning Using Descriptions (SPUD) representation is employed in text generation. The representation developed and used in the input side is very close in content to that of the output side. It is reasonable to provide a translation from the OMAR form of the representation to the PAR form of the representation. Is a translation in the other direction required? The procedure editing capabilities may well operate on a PAR procedure representation and hence, dictate the requirement for the translation from the PAR representation to the OMAR representation. Once again, the similarities of the representations do not preclude this translation. Given the translators, the input and output technologies can be integrated to form a component with capabilities that can be integrated in the larger CAD-based system design environment.

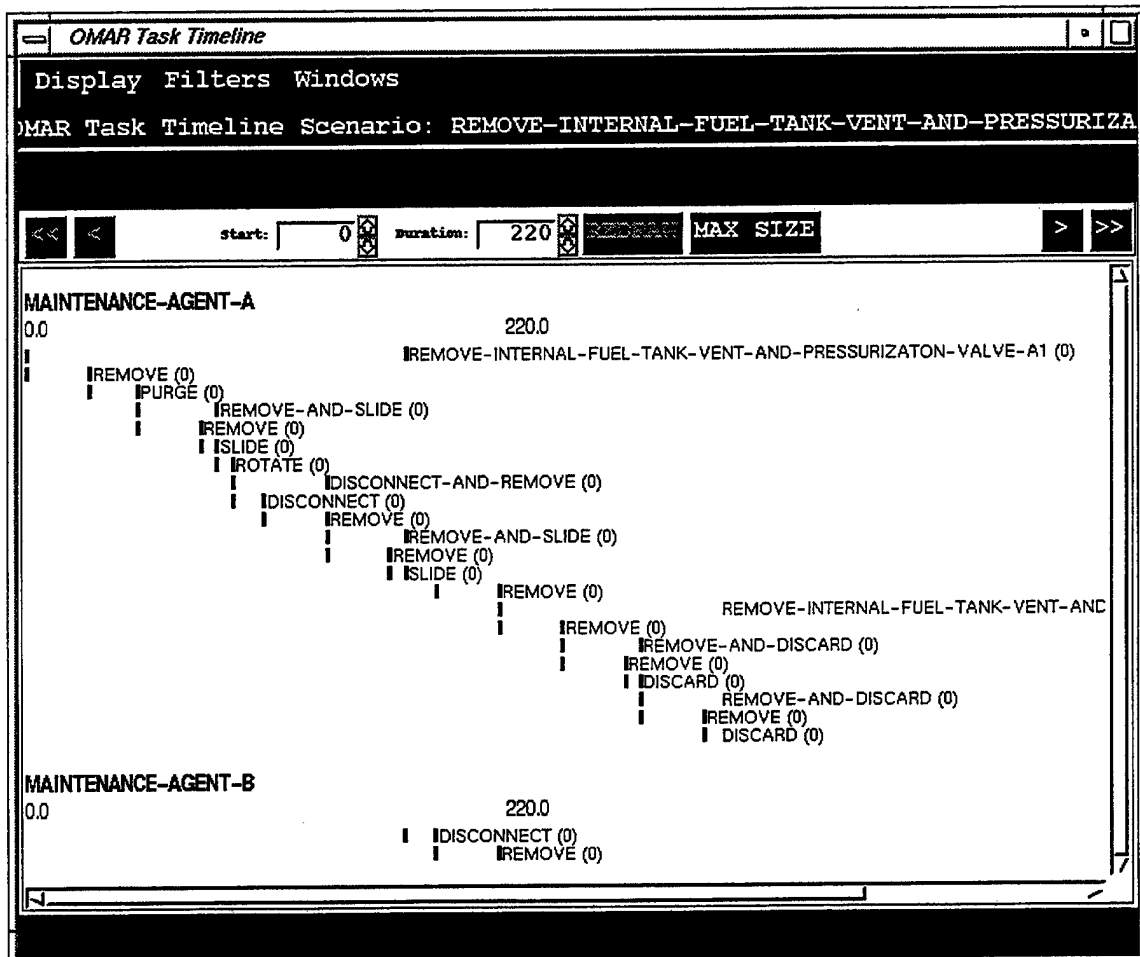


Figure 6. Maintenance Procedure Timeline

2.2 Linguistic Analysis and Language Understanding for AMI

In order to build a system that automates the generation of Technical Orders (T.O.) for maintenance instruction, we first have to consider the knowledge we need to incorporate into the system, and how to obtain that knowledge. For maintenance of complex systems such as the F-16 airplane, CAD databases from the designers and manufacturers can provide necessary information about the types of components involved and their physical relationships to each other. Unfortunately, this is not enough. CAD databases cannot provide experiential or "common sense" knowledge needed to perform maintenance, such as the fact that when a set of bolts are being tightened, it is important to torque them evenly, or that a multimeter should be used to test continuity and voltage of wiring. CAD databases have no information about the proper sequences of actions, which can have an impact on safety as well as efficiency. CAD database tell us which objects are contained in a system, but not how those objects can be manipulated. And since our goal is the automate the generation of T.O.s, it is important to note that CAD databases obviously cannot help resolve issues of style in the generation of T.O.s, the correct level of detail to present to the reader, and the correct tone to use.

Fortunately, a vast amount of domain relevant information exists already in the current T.O.s. The information presented in them is the result of years of experience in general maintenance, as well as direct experience with the systems in question. The exact sequences of actions are explicitly laid out, frequently with notes and cautions to further enhance the safety and efficiency of the procedures. The reader is told which objects are relevant for the task at hand, and exactly how each is to be manipulated. Finally, the T.O.s exhibit a consistent tone and level of detail that is necessary to emulate during automatic generation of new T.O.s.

The issue then is how to take advantage of this wealth of information, how to capture and represent the necessary generalities that are to be exploited in the automated creation and update of future T.O.s. We have investigated the use of linguistic analysis and language understanding techniques to provide a means for extracting and organizing the relevant information from these existing T.O.s. Through linguistic analysis and language understanding, we have developed experimental modules for the automatic capture of procedures and taxonomies for generative planning, for the automatic creation of cases and indices to support Case Based Reasoning (CBR), and for general knowledge acquisition for maintenance procedure planning. We have experimented with test cases, showing how linguistic analysis of T.O.s can assist in the creation of a knowledge database. Our knowledge database was built in SFL (Freeman, 1997).

2.2.1 Automatic Acquisition of Procedures

In order to use linguistic analysis and language understanding for the automatic acquisition of procedures, we had to address the requirements and limitations of such an approach. In order to be used with a hierarchical model of procedures, some knowledge must already be represented in that knowledge. The model must know what kinds of actions exist in the domain, what kinds of objects exist, and what the relationships are between various actions, between various objects, and between actions and objects.

Fortunately, T.O. procedures tend to have very few action types and object types (Badler, Webber, Palmer, Noma, Stone, Rosenzweig, Chopra, Stanley, Dang, Bindiganavale, Chi, Bourne, & DiEugenio, 1997). Figure 7 shows the actions for several procedures involving F-16 fuel tank pressurization valves, and 7 shows the relevant objects. This analysis was sufficient to allow us to create a module that successfully parsed two existing T.O. procedures dealing with the removal of pressurization valves from different fuel tanks.

Action Types
Install x (on y)
Remove x (from y)
Purge x
Disconnect x
Discard x
Slide x (on y)
Rotate x (for y / to y)

Figure 7. Actions in Fuel Tank Pressurization Valve Procedures

Things you can remove	Things you can disconnect
bolts	couplings
washers	tubes
couplings	connectors
access panels	Things you can purge
valves	vent tank
seals	Things you can install
tubes	coupling remover
packing	Things you can discard
Things you can slide	packing
coupler sleeves	Things you can rotate
	elbow

Figure 8. Objects in Fuel Tank Pressurization Valve Procedures

There are also limitations to using linguistic analysis to extract information from existing T.O.s. There are types of information that simply cannot be learned from analyzing existing T.O.s. One such type is what to do with completely new kinds of actions or objects. Obviously, analyzing T.O. procedures for fuel tank pressurization valves will not provide the information necessary to generate procedures for the maintenance of GPS systems. But that is not the goal we are trying to achieve with the linguistic analysis of existing T.O.s. We are attempting to extract and represent the knowledge that is contained in these documents. Necessary information that is not contained in these documents must come from other sources, but that does not lessen the importance of the information we can extract.

Another limitation is that the underlying meaning of actions cannot be extracted from the T.O.s if these underlying meanings are not themselves in the T.O.s (cf. Swartout, 1981). For example, consider the T.O. step, "Purge the vent tank." The steps involved in purging the vent tank, the safety precautions that must be taken, the equipment that must be used, are not given here. The T.O. does not answer these questions, so this information cannot be obtained through linguistic analysis of the T.O.

Again, we should not be surprised that we cannot learn from analyzing the T.O.s information that is not in the T.O.s. If this information is important, if we want it to be represented in our model, we have to find other sources for this information. If the information is not important, if we do not need it in our representation, then it doesn't matter that we can not obtain it from analyzing existing T.O.s. In the automated generation of T.O.s, we have to identify the level of detail that is appropriate to present in the generated T.O.s.

One way to determine this is to look at the level of detail provided in existing T.O.s. To continue our example, the T.O.s never describe exactly what is meant by "discarding" an object. It is assumed that the technician reading the T.O. is intelligent enough to know how to discard an object appropriately. It would be a mistake in writing a T.O. to expand on this. Linguistic analysis of existing T.O.s can give us valuable information concerning

the levels of detail that are correct when generating new T.O.s. Providing too detailed an explanation makes the new T.O. tedious, obscures the important information to be conveyed to the technician, and ultimately makes the T.O. unusable. Providing an insufficient level of detail causes the technician to receive less information than he needs to successfully complete his task, and is equally unacceptable. Therefore, it is important to obtain the correct level of detail, and linguistic analysis of existing T.O.s can help us achieve this goal.

Finally, and closely related, is the limitation of linguistic analysis concerning implicit information. For example, consider the T.O. step, "Rotate elbow to provide clearance around valve." What is "clearance around valve"? Nowhere in the T.O. is this explained, so analysis of the T.O. cannot help you discover it. If this information is important, other sources will have to be found to provide the answers. In spite of these limitations, analysis of existing T.O.s provides invaluable aid in determining the level of detail appropriate in the generation of new T.O.s.

Linguistic analysis of existing T.O.s can also reveal particular sequences of actions. If there is a particular order in which certain actions occur, analysis of existing T.O.s will capture this knowledge so that it can be represented in the model. For example, packings are usually lubricated before they are installed. Analysis of the T.O.s would reveal this general rule and allow it to be represented in the model.

Finally, linguistic analysis can reveal which ways particular objects can be manipulated. Returning to Figure 8, we see that the objects are organized in the table by the actions that can be performed on them. This organization comes from analysis of the T.O.s for fuel tank pressurization valves.

2.2.2 Text Analysis for Knowledge Acquisition

Our text analysis module uses the parser Sparser (McDonald, 1992). Sparser is a bottom up chart parser¹ which uses a semantic phrase structure grammar. The use of a semantic phrase structure grammar is central to our approach. Semantic grammars break up the text into semantic categories, such as PROCEDURE-STEP or REMOVABLE-OBJECT, rather than syntactic ones, such as VERB or NOUN-PHRASE. The advantage of this approach is that the resulting parse structure contains the hierarchies and relationships we want in our procedural representation model instead of containing syntactic information that would have to be translated in nontrivial ways to obtain the hierarchies we want to represent.

Traditional syntactic parsing has important limitations. They can provide information about sentence structure, but not about the meaning of the sentence. And since syntactic grammars have not been effectively developed for text units larger than sentences, a syntactic parsing approach cannot yield any information about the relationships among the steps of a T.O.

As an example, consider the syntactic parse tree in Figure 9.

¹ See [Winograd 1983] for an overview of chart parsers.

This figure shows a typical syntactic parse for the sentence "Remove access panel 3405." This parse does provide useful information. It shows us that the word "remove" is being used as a verb, followed by a noun phrase consisting of the compound noun "access panel 3405." This sentence structure is important to have when generating new text in order to achieve the correct style. But, unfortunately, this syntactic structure does not capture any similarities or differences in meaning between sentences of identical syntactic structure, as Figure 10 through Figure 12 illustrate.

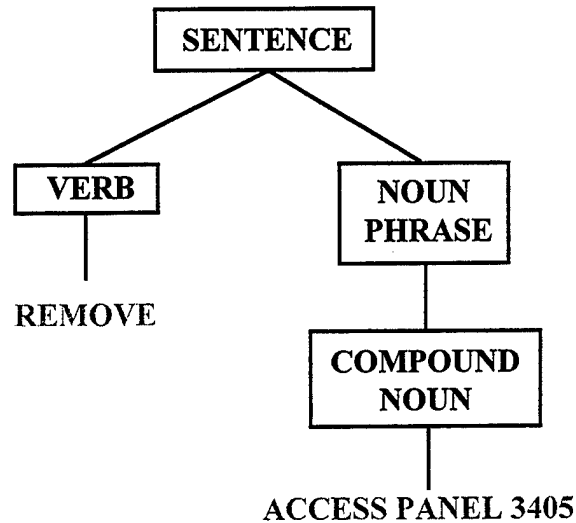


Figure 9. Syntactic Parse of "Remove access panel 3405."

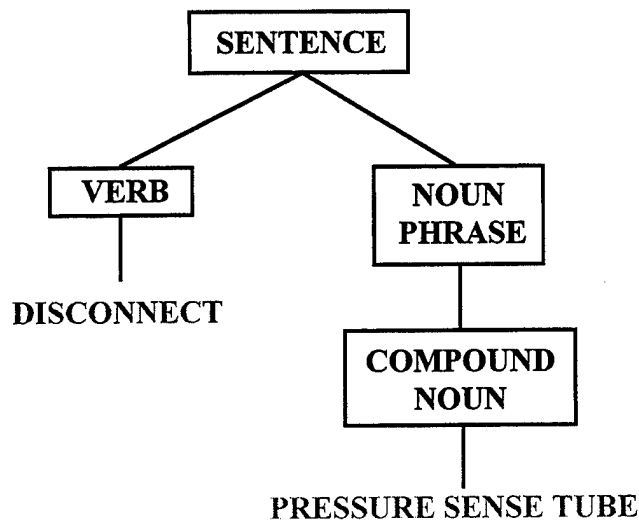


Figure 10. Syntactic Parse of "Disconnect pressure sense tube."

All these sentences have the same syntactic structure: a verb followed by a noun phrase consisting solely of a noun or a compound noun. However, these parses do not capture the fact that the actions described by these verbs are all very different, as are the objects these actions are being performed on.

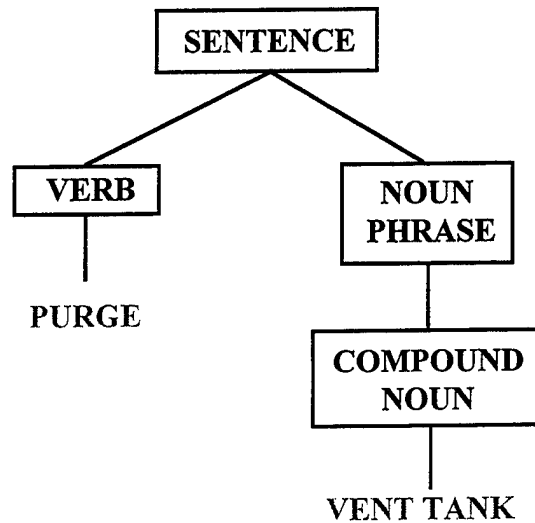


Figure 11. Syntactic Parse of "Purge vent tank."

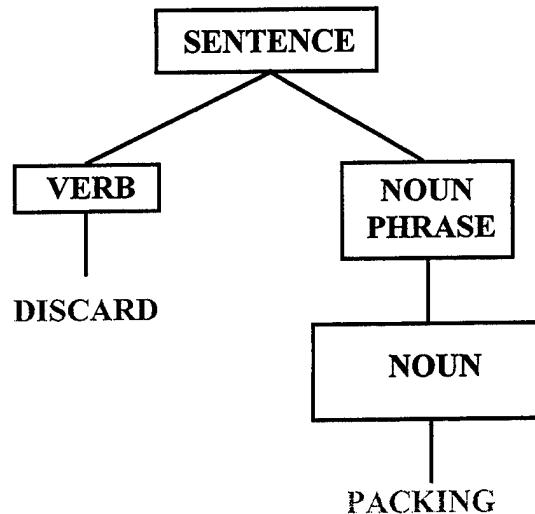


Figure 12. Syntactic Parse of "Discard packing."

Alternatively, parsing with a semantic grammar instead of a syntactic one allows us to use semantic categories that correspond directly to classes for actions and objects needed in the knowledge base for maintenance procedures.

Figure 13 shows us such a semantic parse. Here instead of being told we have a sentence consisting of a verb phrase followed by a noun phrase, we are told that we have a STEP COMMAND consisting of a REMOVE COMMAND. The REMOVE COMMAND is composed of the word "remove" followed by a REMOVABLE OBJECT. The REMOVABLE OBJECT consists of an F-16 OBJECT, which consists of the phrase "access panel 3405." These are the very categories and restrictions we need to represent in the knowledge base of maintenance procedures.

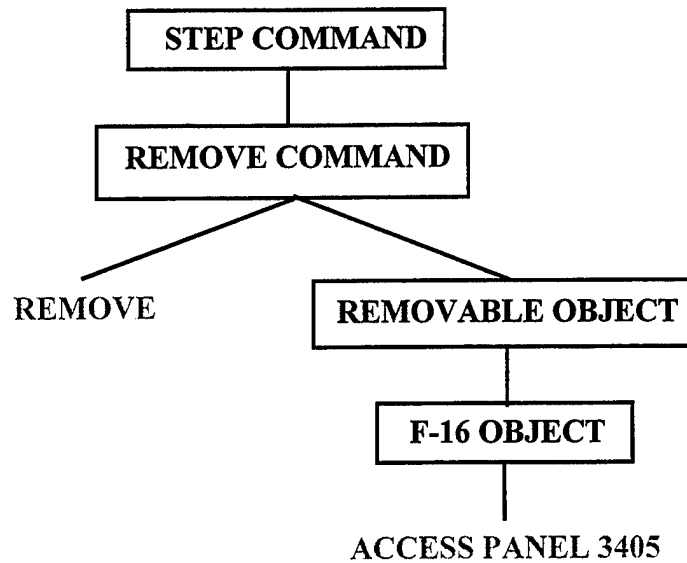


Figure 13. Semantic Parse of "Remove access panel 3405."

The rules for defining these relationships are straightforward in Sparser. Figure 14 shows the few rules necessary to obtain the parse in Figure 13.

```

(DEF-CFR STEP-COMMAND (REMOVE-COMMAND))
(DEF-CFR REMOVE-COMMAND ("remove" REMOVABLE-OBJECT))
(DEF-CFR REMOVABLE-OBJECT (F16-OBJECT))
(DEF-CFR F16-OBJECT ("access panel 3405"))
  
```

Figure 14. Semantic Rules to Parse "Remove access panel 3405."

Additionally, we can define semantic categories and write grammar rules that will allow semantic parses for text units larger than sentences, as Figure 15 illustrates.

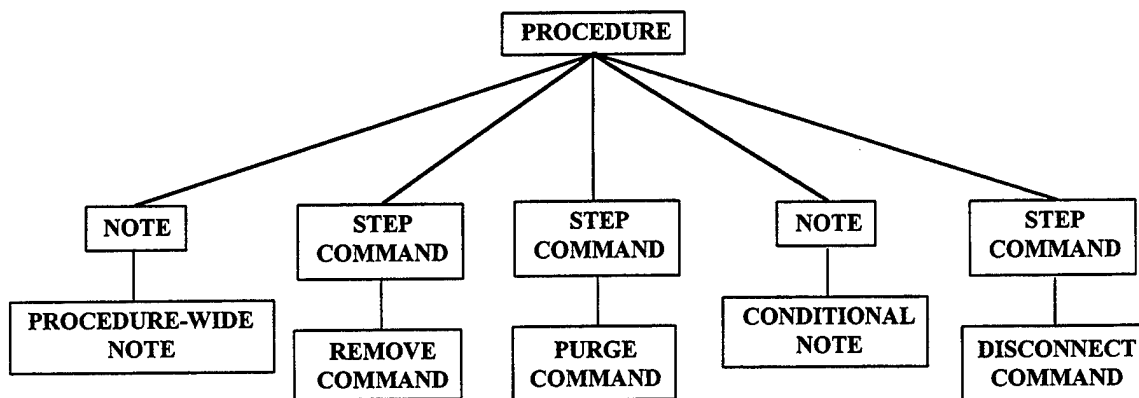


Figure 15. Semantic Hierarchy for T.O. Procedure

Capturing such organization yields not only necessary information on how each sentence should be organized when generating a T.O., but also how the various sentences are organized into sections, and how those sections can be organized with each other.

2.2.3 Semantic Parsing for Generative Planning

Using the Sparser system with a semantic grammar feeds directly into our generative planning module. As we have seen, the semantic categories of the rules suggest a natural hierarchy for knowledge model classes. For example, we have seen that access panel 3405 is an F-16 OBJECT, and that a STEP COMMAND might consist of a REMOVE COMMAND, a PURGE COMMAND, or a DISCONNECT COMMAND.

The semantic categories of the parse can also be used to suggest restrictions and requirements on the objects of procedures. For example, the object of a REMOVE COMMAND must be a REMOVABLE object. Furthermore, since the T.O. has an example of a REMOVE COMMAND involving access panel 3405, that panel must be a REMOVABLE OBJECT.

It should be noted here that while the semantic categories can suggest hierarchical classes for the knowledge database, semantic parsing of the T.O.s cannot be relied upon solely as the source for these classes. The classes must capture more knowledge and generalities than are exhibited explicitly in the T.O.s, and as we noted above, information that is not explicitly given in the T.O.s cannot be obtained only through analysis of the T.O.s. For example, if there are useful generalities needed for generative planning, such as the fact that REMOVE and DISCONNECT are both actions to DISENGAGE an object from another, or that BOLT, SCREW, and CONNECTOR are all types of FASTENER, this knowledge cannot be obtained from analysis of existing T.O.s. However, if such a hierarchy is in place, semantic analysis could populate the database by creating database objects of the appropriate classes, confirming the class hierarchy (or alternatively, demonstrating mistakes in the class hierarchy during development).

Sparser has mechanisms that allow its rules to create or modify knowledge model objects. Each rule has an optional "referent" field associated with it. This referent field contains a function to build or add to the associated referent object in the knowledge model. The rule defining "access panel 3405" as an F16-OBJECT has a function that retrieves the knowledge database object representing access panel 3405. Similarly, the rule that defines a REMOVABLE OBJECT as an F-16 OBJECT marks the object retrieved from the database (in this case the object representing access panel 3405) as REMOVABLE.

This example is simple enough, because there is only one access panel 3405 on an F-16. But what happens when the reference is not so straightforward? Take for example the step, "Remove coupling and slide sleeve on elbow." There are probably thousands of couplings, sleeves, and elbows on an F-16. Correspondingly, our knowledge database will have thousands of objects representing couplings, sleeves, and elbows. How can we possibly know which ones to retrieve for this particular step? We use the notion of context to help disambiguate these references. That is, if a reference is not unique in a given F-16, we search the text for a more restrictive domain in which the reference is unique. The reference "access panel 3405" is unique because there is only one such access panel on a plane. The references "coupling," "sleeve" and "elbow" are not unique, however.

The first restrictive context we use is that of the procedure itself. The example we are using comes from procedure 2-14-1. Removal of Internal Fuel Tank Vent and Pressurization Valve. Therefore, we are only interested in objects associated with the internal fuel tank vent and pressurization valve. Looking through the knowledge database, we find two couplings and sleeves associated with this valve, but only one elbow. Therefore the reference "elbow" must be to the object representing the elbow for this valve.

Now we still have to disambiguate the other two references. We can further restrict the context to objects associated with the elbow we have disambiguated. Looking through our database again, we find only one coupling and sleeve associated with the elbow associated with the internal fuel tank vent and pressurization valve. Through subsequent refinement of our context, we are able to disambiguate all references in the step, and retrieve the correct database objects associated with these references.

2.2.4 Semantic Parsing for Case Based Reasoning

Just as the semantic grammar contributed to building the generative planner, it is used to help build the Case Based Reasoning module. The semantic categories suggest a natural hierarchy of processes and objects that can be used to identify major indices for CBR. Additionally, the process and object hierarchies provide the basis for creating abstractions and generalizations of the cases.

Again we exploit the Sparses rule's referent field to populate the case database once the CBR indices have been identified. For example, after analyzing the procedure 2-14-1, the CBR database would have several cases representing REMOVE actions. They would be indexed on REMOVE, and inspection of each of these cases would reveal that in each case the object being removed was a REMOVABLE OBJECT. Therefore, in generating a new REMOVE case, the system would ensure that the object to be removed was marked REMOVABLE.

Just as with generative planning, we need to assure that the correct CBR database objects are being indexed in each case. Therefore, our mechanism for disambiguating references by subsequently refined context must be used for CBR case generation, too.

2.2.5 Concluding Remarks on Linguistic Analysis and Language Understanding

This section has shown that linguistic analysis and language understanding play an initial role in the knowledge acquisition phases of developing an automated maintenance instruction system. Analysis of existing T.O.s is a critical step in the creation of a knowledge database necessary for generative planning and for case based reasoning. A language understanding module can not only perform this analysis, suggesting class hierarchies for procedures and actions necessary for both the generative planning and the CBR approaches, but it can automatically populate this database.

Our use of semantic grammar rules allows us to capture directly in our rules the generalities we wish to extract from the T.O.s. The semantic categories correspond to the hierarchical classes for generative planning, as well as the indices for CBR. Additionally,

the semantic rules allow us to analyze the text at units larger than the sentence, in order to capture useful generalities exhibited at different levels of the text.

2.3 Planning Technology for AMI

The objective of the AMI research is to create automation technology that can reduce the cost of producing and revising maintenance instructions. Candidate tools for automating these publications and streamlining their associated processes include: generative planning, to support the *derivation* of maintenance procedures; and case-based planning to support the *re-use* of maintenance procedures.

Our vision of AMI includes a user interface that enables an author to graphically build maintenance procedures using libraries of previously built procedures and procedure templates (schemas). In this vision of AMI, the author will also be able to build and revise a procedure through access to many of the types of tools that are currently available to Technical Order authors, including:

- a CAD database,
- other procedures that have been authored in the past (a case repository),
- a library of procedure templates (schemas) that contain process flows for building certain types of procedures,
- access to cautions and warnings,
- graphical tools that support 2-D and 3-D visualization and manipulation,
- text generation and text understanding tools,
- simulation tools.

The CAD database is of particular value to the AMI author, who should be able to access and copy elements from a CAD database for use in the current work space. In our AMI vision, access to the CAD database is facilitated through cross-references of the type typically employed by case-based planning systems. In addition, the author is provided with methods that support:

- manipulation of drawings
- labeling of drawings
- composition/decomposition of elements, e.g., a fastener
- development and maintenance of an ontology of components and their attributes.

In order to realize our vision of AMI, the computer *representation* of aircraft maintenance procedures is essential. For example, we believe that part of the authoring process involves filling in a computer representation such as a schema (which is like a template that provides *planning* guidance for building something). We envision different schema for different types of procedures, e.g., remove, replace, cautions, warnings, partial order constraints, and that these schemas would be available as guidance to the generative planner or stored in a case-base for re-use by a case-based planner.

2.3.1 Using Generative Planning to Promote Automation

As we have previously discussed, the linguistic features of technical manual writing—a prescribed format, a terse and repetitive style, limited vocabularies, and simple grammar—lend themselves to automated parsing and interpretation. Even a casual perusal of the technical manuals reveals striking similarities among maintenance procedures, not just in the linguistic conventions they employ, but also in the underlying logical organization of the actions they describe. This section focuses on techniques for identifying principles underlying the logical sequence of steps seen in maintenance procedures. Our focus on existing technical manuals and authoring practices is a methodology for revealing the embedded common sense and doctrine underlying sound maintenance instruction. Our goal is to represent this common sense and doctrine in a knowledge base of object descriptions, plans, constraints, and rules of thumb, freed from the literal context in which it currently appears—text and pictures on the printed page (whether printed or electronic).

We firmly believe that the common structural (i.e., logical, not just linguistic) features of maintenance procedures can be captured in a plan representation language and used to accelerate the authoring process. In this vision, the author guides and critiques the procedure generation process, then polishes the result by filling in details and specifying idiosyncratic aspects of the procedure. The routine decisions about what steps are required and what order they appear are made by a process of expanding a general plan in accordance with the particular knowledge of aircraft components and general common-sense knowledge about maintenance practices. A goal of AMI is to maximize the extent to which information necessary to guide the plan expansion process can be accessed or inferred from existing CAD/CAE databases.² To this end we looked in detail at several representative procedures—selected from a corpus of F-16 fuel distribution system technical manuals—with three parallel thoughts in mind:

- 1) What types of higher level structure are apparent in the procedures?
- 2) What types of detailed information would be needed to fully specify the procedure, starting from a general outline?
- 3) What types of common-sense knowledge are apparent in the procedures?

For purposes of exposition, we will focus on a family of procedures for the removal and installation of fuel system components:

2-14. Internal and External Fuel Tank Vent and Pressurization Valves, 2821FV2 and 2821FV3, Removal, Installation, and Checkout

A representative page of these procedures can be seen in Figure 3. Our methodology was to examine each pair of procedures for similarities and differences; to compare and

²We envision a collection of software agents (specialists) that mediate between the demands of the plan expansion process and the database content. Some agents simply embed the ability to query the databases and manage the results; at the other extreme are agents that embed specialized reasoning abilities; e.g., a human form model to analyze component accessibility.

contrast the removal procedures for two different components and the installation and removal procedures for the same component. The possible comparisons can be seen in Figure 15.

	Internal	External
Remove	2-14-1	2-14-2
Install	2-14-4	2-14-3

Figure 16. Pair-wise Comparison of Fuel Tank Vent and Pressurization Valve Maintenance Procedure

2.3.1.1 Removal

To remove a component:

1. locate it
2. disable it (turn off, drain, depressurize,...)
3. remove obstacles
4. remove connections (electrical, fluid, mechanical (support))
5. remove it

This abstract form of a removal procedure captures the essential elements of both 2-14-1 and 2-14-2, as it does all the other removal procedures in the corpus. It defines necessary general actions and the order in which they should be performed. Differences between the procedures reflect variations in the expansions of the individual steps in this procedure—e.g., there are more connections on the external tank valve—but the overall order remains intact: For example, in the case of the internal tank valve (2-14-1) the expansion results in the following sequence:

- (1) remove access panel (step 1)
- (2) purge vent tank (step 2)
- (3) rotate elbow (steps 3-4)
- (4) disconnect the tubes, the electrical connector, the mounting bracket (steps 5-9)
- (5) remove the valve (step 10-12)

What knowledge is required to render this expansion? Three types of knowledge form the core of a knowledge representation for physical systems:

Classification—the assignment of components to types or classes.

Part-Whole—the aggregation of components into groups.

Connection—the physical (and functional) attachment between components that defines how behaviors influence one another.

These and other types of knowledge reveal themselves in the steps as follows:

Location: Step 1 requires knowing the valve's location and its primary means of access (door, panel).

1. (A) Remove access panel 3405.

System affiliation: Step 2 requires knowing the safety procedures associated with the system of which the valve is a part (the fuel distribution system). The functional relation of a component to its system is a key to recalling safety issues, cautions, and notes.

2. (A) Purge vent tank.

Obstruction: From our examination of the corpus of procedures, the predominate source of constraints on assembly and disassembly is the need to access, manipulate, and extract components and the tools needed to free them. The spatial reasoning required to recursively plan actions is central to the AMI vision.

Clearance: Step 4 requires knowing the difference between removing and moving out of the way (rotating, in this case). The potential for this type of movement is a property of the connectors; namely, the degrees of freedom at the joint.

4. (A) Rotate elbow to provide clearance around valve.

Mechanism: Types of objects convey information about the actions needed to manipulate them; e.g., the type of connector dictates the actions in the Step 6.

6. (A) Remove coupling and slide sleeve on pressurization tube.

Parts: The decomposition of components into their parts enables knowing about the packings that are contained within the connectors used in the fuel system.

11. (A) Remove and discard four packings.

Properties: Step 11 illustrates a particular property of these particular packings: they are not reusable.

Recombination: Step 11 also illustrates another property of these procedures: the aggregation of like actions arising into a single compound step. The four packings arise from two prior disconnect actions, but are recombined into a single step of the procedure.

Tools: Actions on some components require specialized tools. The single notional disconnect step gives rise to three steps describing the installation and removal of the tool required in this circumstance.

12. (A) Install *coupling remover* on external vent and pressurization valve and external tank and pressure tube.
13. (A) Slide sleeve on external tank vent and pressure tube forward by moving *coupling remover* lever aft.
14. (A) Remove *coupling remover* from tank.

2.3.1.2 Installation

Further insight into the generic qualities of maintenance procedures is revealed by comparing installation and removal procedures for a single component. Common sense tells us that they are rough inverses of each other. However, interesting differences can be seen by comparing corresponding steps from procedures 2-14-1 and 2-14-4 as indicated by the symbol ② below.

Level of Detail: Installation requires more detail than removal for the same operation. For example, the designation of any supplies, how hard to tighten bolts, and the number of fasteners become issues during installation but are not relevant to removal.

12. (A) Remove union and packing. Discard packing.

② 1. (A) Lubricate and install packing (M25988/1-904) and union. Torque to 72-78 inch-pounds.

1. (A) Remove access panel 3405.

② 12. (A) Install access panel 3405 using 27 bolts.

Subprocedures: A single logical action can expand into several detailed actions in a technical manual. This example shows the fine points of mounting a component that has two attachment points; it reveals the common-sense skills of a technician who will first attach one mount, leaving it untightened, then attach the other before tightening both mounts.

9. (A,B) Remove four bolts and four washers securing valve to bulkhead.

② 5. (A,B) Position valve and seal on bulkhead and install four bolts and four washers. Do not torque.

6. (A) Align valve on bracket and install bolt and washer. Torque to 40-60 inch-pounds.

7. (B) Torque four bolts to 110-140 inch-pounds.

Personnel: How many technicians are needed to carry out an action. The previous example calls for two people, presumably because of the awkwardness of positioning a part and simultaneously starting the bolts.

Consistent References: The two procedures exhibit minor inconsistencies in the names given to components. This is one of the ways in which AMI can improve the quality of technical manuals—removing ambiguity and potential sources of confusion—by insuring that consistent names and references are used. In this example, the same part (sense tube) is referenced in two different ways in two procedures; we speculate that in one case the name was derived from its *role* (i.e., pressure) and in the other from its *owner* (i.e., valve).

5. (A) Disconnect and remove pressure sense tube.

② 8. (A) Install valve sense tube. Torque to 72-78 inch-pounds.

2.3.1.3 Authoring as Plan Expansion

Although these examples are drawn from only four procedures, the logical regularities and the types of implicit knowledge represented in them are reflected in all other procedures we have examined in the corpus. In building an AMI authoring environment, one would supplement this second-hand source with the first-hand knowledge of subject matter experts and authors in order to capture other conventions and system-specific knowledge needed to guide the plan expansion process.

We envision an authoring environment as providing the author with a collection of specialists. Expertise resides in the class hierarchy that describes component types and the structural knowledge that describes containment (part-whole) and connectivity relationships. In the plan expansion, a spatial reasoner both introduces new steps and imposes a (partial) ordering on steps to account for the removal of obstacles. Another specialized reasoner captures the conventions for dealing with *fastening*: the notion of attachment points, the manipulations required to install and remove fasteners, tools requirements, fixtures, personnel requirements, parameters (torque, e.g.), and subprocedure templates for attaching and securing multiple fasteners. A related reasoner focuses on *connectors*: manipulations, tools, seals (reusable and otherwise), properties of the contents of the flow path being joined or disconnected, and so on.

Plan expansion is an iterative process of generating steps, adjusting the order of the expanded steps, and in some cases re-aggregating steps into new groupings. For example, in 2-14-1 two disconnects had packings to be removed (and discarded) as part of the disconnection subprocedure. These steps were migrated to the end of the procedure and collected into a single compound statement (Step 11). This could have been generated by an accessibility imperative (the packings are easier to reach after the valve has been extracted from the aircraft) or from a convention associated with the class of the connection.

In examining maintenance procedures one notices that not all constraints on the order of steps are physical imperatives. One sees institutional constraints that embody safety considerations and good work practices (removing potential hazards, keeping control of the situation) and convenience constraints that serve as memory aids (natural groupings of elements or sequences of actions). When plans are expanded, constraints on the order of steps at a general level become reattached to some or all of the constituent steps. For example, in 2-14-1, the general physical constraint on the disconnect actions (that they occur before the valve is removed), when the procedure is expanded, attach to the actual decoupling actions, not the subsidiary steps of removing the packing, which migrate to the end of the procedure as described above.

These examples suggest the types of general system and planning knowledge that would be the focus of the knowledge acquisition for an authoring environment and the bridges that need to be built to allow the CAD/CAE repositories to feed the authoring process. The power of the generative approach is the robustness of the resulting procedure, its explicit ties to general principles and guidelines, and its ability to free the author from the repetitive aspects of procedure specification. The burden of the generative approach is the

knowledge acquisition effort to capture and formalize the regularities that form the foundation for maintenance procedure specification in our approach.

2.3.2 Using Case-based Planning to Promote Automation

Case-based reasoning (CBR) systems capitalize on the observation that human problem solvers often derive all or part of a solution to a current problem from all or part of a solution to a problem that they “remember” encountering and solving in the past. Case-based reasoning systems “remind” a user of their past experience at a time when they are faced with a similar problem solving situation. CBR systems have been found to be particularly useful when: (a) previous experience is available to improve the problem solution and avoid past failures (Riesbeck & Shank, 1989), (b) there is incomplete information about the current problem, or (c) the current problem must be quickly solved. When there is a store of relevant experience, CBR can provide an effective approach to drawing on that experience to solve current problems. Technical manual authoring is just such a case. Existing Air Force T.O.s form a huge repository of information relevant to the authoring of new T.O.s.

CBR has been successfully used in a wide variety of applications, including: help desk support (Mulvehill & Christopher, 1991; Mark, Simoudis, & Hinkle, 1997), training (Shank, 1997); cooking (Hammond, 1986), cardiac diagnostics (Koton, 1988), planning (Carbonell, 1986; Mulvehill, 1995; Veloso, 1994), and thunderstorm prediction (Nicholson & Mulvehill, 1990). In each of these applications, prototypical cases were obtained from domain experts, and when stored as cases in a case-base proved useful in both suggesting solutions and in warning of possible problems that might arise (Leake, 1997). Further, once built, CBR systems have generally been well received by users because: (a) the knowledge base (case-base) consists of experiences that are familiar to the user, (b) the case-base is automatically updated with new experience, and (c) training associated with system usage is reduced due to familiarity of the user with the contents of the case-base.

When CBR is used to build a planning system, the resulting planning system is referred to as a case-based planner (CBP). CBP is used by systems like Prodigy-Analogy (Veloso, 1994) and CHEF (Hammond, 1986) to make use of past plans in quickly building solutions to current problems. Prodigy-Analogy has been used to derive the best transportation route between two locations. CHEF prepares plans for cooking meals for people with special dietary restrictions.

Comparisons between case-based planning systems and generative planning systems indicate that a plan can be more quickly constructed when past plans are reused than when they are produced generatively (Koton, 1988, Veloso, 1994). However, studies also indicate that reusing past plans to build new plans can result in plans that contain over-generalized solutions (e.g., when hungry, always eat Chinese food).

CBR systems solve new problems by using case indices to search for the best match between the description (a set of indices) of the new problem and descriptions of problems solved in the past. The main components of a CBR system include: a case

library; a case retriever and a case storer that both use an index generator to extract key features from cases; a case filtering mechanism; a rule-based case adapter; and a case generalizer to help in the formation of templates from specific examples. Indices used for storage and retrieval are based on a combination of goals, key aspects of the situation in which the procedure must be applied, and additional information such as the priorities of the individual objectives, the amount of time available for task completion, and the quality of information required.

As the size of the case library grows, more example cases will be found than are needed for new procedure formation. A case filtering mechanism can be used to provide more careful matching of retrieved cases with the current situation, in effect ranking the cases so that only the most relevant options are presented. Another way to distinguish important variants as alternative plan strategies is to build templates that represent these alternatives. The case generalizer examines cases similar to the one being stored and identifies the elements that vary among the similar cases, and develops templates for each major alternative with specific knowledge and heuristics that are determined to be necessary to adapt the templates to new requirements (Burstein, 1994).

2.3.2.1 Related Case Based Reasoning Research and Systems

We believe that while authoring T.O.s, the author is primarily engaged in a design task and requires access to a variety of information including: CAD drawings, related T.O.s, and T.O. notes, cautions, and warnings. CBR has been used successfully in the design and development of several design systems. Some examples include: JULIA (Leake, 1997) a system that designs meals; CYCLOPS (Navinchandra, 1988) a system that uses CBR for landscape design; KRITIK and KRITIK-2 (Goel & Chandrasekaran, 1989; Stroulia, Shankar, Goel, & Penberthy, 1992) that combine case-based with model-based reasoning to support the design of small mechanical and electrical devices; and CADET, (Sycara, Guttal, Koning, Narasimhan, & Navinchandra, 1991) a system that employs a combination of case-based and causal knowledge to do design. In addition, the recent work of Voss, Grather, and Schmidt (1997) focused on developing methods for using CAD drawings to support case-based design. In particular, the focus of this work was to enable a user to "copy" and "paste" operations from a CAD design library into active designs. The users are architects who tend to "reuse" old designs (cases) for extracting design ideas. This work is an example of the already existing link between CAD design and CBR.

Another related environment that could contribute significantly to the development of a system that automates maintenance instruction is work on case-based Help Desks. Help Desks provide links to reference diagrams and other documentation that support the generation and revision of a product. They may also provide reference to other information, for example, models of how things work that enable a user to better understand the problem. While standard Help Desks provide HTML links to reference diagrams and other domain information, case-based Help Desks can provide additional information such as advice, for example, accessing those parts of an aircraft that are known to involve exposure to spilled fuel. Mark et al. (1997) have found that CBR

enhances the effectiveness of a Help Desk by providing mechanisms that access warnings that are associated with specific operations. For example, when performing a certain task, an HTML link would also reference advice like the following:

To avoid fire and explosive hazards, spilled fuel shall be cleaned up immediately. If fuel spillage occurs on surface of aircraft, area shall be checked to determine if fuel has impregnated insulating blankets or duct insulation. If evidence of impregnation exists, insulating blankets or duct insulation shall be replaced prior to engine operation.

2.3.2.2 A CBR Experiment

In order to determine the relevance of using CBR for automated maintenance instruction, we conducted an experiment using the ForMAT CBR system (Mulvehill, 1995). In this experiment, the following two maintenance procedures were represented as cases (see Figure 17 for an example):

- 2-14-1 Removal of Internal Fuel Tank Vent and Pressurization Valve
- 2-14-4 Installation of Internal Fuel Tank Vent and Pressurization Valve.

Since we did not want to make any changes to the CBR system being used, some of the indices that are associated with all of the case-bases made with this system, e.g., modifier, modification-date were automatically inherited by the cases. In addition, although the CBR system used includes methods that can be used for automatic indexing, the two cases used in this experiment were manually indexed in order to support search and retrieval experiments and in order to experiment with procedure comparison. During the experiment, a similarity was identified between the indexing techniques employed by SPARSER and the indexing employed by the CBR system. Based on this observation, we believe that the SPARSER index could be used to support automatic case indexing.

Several successful search and retrieval experiments were conducted in order to validate the usage of the index terms for search and to determine search precision. No tests were made for search speed since the case-base created for this experiment was composed of only two procedures.

BASIC Case REPORT: *** (141 - 01) *******

TITLE: 2-14-1 Removal of internal fuel tank vent and pressurization

FEATURES (Indices): VALUES

CREATION-DATE: Dec-6-1997

AUTHOR: AMM

DISCONNECT-COMMAND: DISCONNECT

GOAL: PROVIDE-CLEARANCE

GOAL: PURGE-TANK

GOAL: REMOVE-ACCESS-PANEL

GOAL: REMOVE-COUPLING

ID: |141|

Item#: |3405|

MODIFICATION-DATE: Dec-6-1997

MODIFIER: AMM

PROCEDURE-STEP: 2-14-1-5A

REMOVABLE-OBJECT: ACCESS-PANEL

REMOVABLE-OBJECT: ELBOW-COUPLING

REMOVABLE-OBJECT: ELBOW-SLIDE-SLEEVE

REMOVABLE-OBJECT: PRESSURE-SENSE-TUBE

REMOVE-COMMAND: REMOVE

STEP-NUMBER: |5|

STEP-PERSONNEL: A

TASK: DISCONNECT-AND-REMOVE-PRESSURE-SENSE-TUBE

TASK: PURGE-VENT-TANK

TASK: REMOVE-ACCESS-PANEL

TASK: REMOVE-ELBOW-COUPLING

TASK: REMOVE-ELBOW-SLIDE-SLEEVE

STEP#S:

STEP917	: 21415 ::REMOVE ACCESS PANEL 3405 (GENER...
STEP918	: 21411 ::DISCONNECT AND REMOVE PRESSURE...
STEP919	: 21412 ::PURGE VENT TANK (T. O. 1-1-3)...
STEP920	: 21413 ::REMOVE COUPLING AND SLIDE SLEEVE...

Figure 17. Basic Case

2.3.2.3 Comparing Plan Components Versus Entire Plans

As procedures are modified by the author, tracking the changes over time becomes tedious. The CBR system used for this experiment provides several reports that are useful for comparing cases (i.e., procedures in this example). Figure 18 displays part of the results from the report mechanism that was used to compare the starting 2-14-1 procedure (214-01) with the revised 2-14-1 procedure (141-01) as it was modified by the author (modifier = amm). In this report, the CBR system indicates that one step (STEPA19) was replaced by four steps (STEPGDC, STEPGDP, STEPH19, and STEPH7B) in the revised procedure.

Comparison between (214-01) and (141-01)	
(214-01): 2-14-1 Starting procedure.....	
(141-01): 2-14-1 -REVISED- Removal of internal fuel tank vent.....	
Number of Step#s in (214-01): 15	
Number of Step#s in (141-01): 12	
Step#s in both (214-01) and (141-01): 11	
STEP917	: DISCONNECT ELECTRICAL CONNECTOR.....
STEP918	: DISCONNECT AND REMOVE PRESSURE.....
STEP919	: SLIDE SLEEVE ON EXTERNAL TANK VENT.....
STEP920	: REMOVE COUPLING REMOVER FROM TANK.....
STEPA17	: REMOVE ACCESS PANEL 3405 (GENER.....
STEPA18	: ROTATE ELBOW TO PROVIDE CLEARANCE.....
STEPA7B	: REMOVE COUPLING AND SLIDE SLEEVE.....
STEPB17	: REMOVE 4 BOLTS AND 4 WASHERS.....
STEPB19	: REMOVE VALVE AND SEAL FROM TANK.....
STEP17	: REMOVE UNION AND PACKING.....
STEP7B	: RECEIVE AND DISCARD 4 PACKAGES.....
Step#s in (214-01) but not (141-01): 4	
STEPGDC	: REMOVE COUPLING FROM EXTERNAL TANK.....
STEPGDP	: INSTALL COUPLING REMOVER.....
STEPH19	: PURGE VENT TANK (T. O. 1-1-3)....
STEPH7B	: REMOVE BOLT AND WASHER FROM.....
Step#s in (141-01) but not (214-01): 1	
STEPA19	: REMOVE COUPLING AND SLIDE SLEEVE.....

Figure 18. Case Comparison Report

This type of report information is automatically provided by the CBR system and has been found (in the application of this system to another domain) to be useful for tracking changes to procedures over time. This capability has proved especially useful as case-base size increases.

2.3.2.4 Suggestions for CBR Usage

Although automation of procedure generation is necessary for speed, review by human beings is still necessary. Human beings will accept this division of labor only if the "first-draft" generated by the automated tools is consistent with their expectations.

We believe that a system that supports the automation of maintenance instructions could benefit from a combination of a generative planner with a CBR system. In this design, the generative planner would be used to support the initial generation of procedures, and the CBR component would be used to provide "reminders" to the author and to enable the author to build a plan by "cutting" from previous similar procedures and "pasting" the

relevant aspects of those procedures into the new procedure. We have identified other CBR tool development areas (e.g., CAD system design, supporting training, and Help Desks) that should also be considered as a source of future automation support system design.

2.3.3 Combining CBR and Generative Planners

CBR systems are only useful when they have past experience (stored as cases in a case-base) to use in problem solving. When this is not the case, or when the modification of a solution to a past problem is extremely complicated or too tedious, it is often better to rely on a generative planner to produce the plan or problem solution.

The benefit of combining CBR with generative planners is that each generated plan (or problem solution from the generative planner) is stored as a case in the case-base and can be retrieved for use by a human or by the generative planner in solving similar problems. Past cases can be used to "remind" the human planner and the generative planner of failed or successful problem solutions.

The Prodigy-Analogy system (Veloso, 1994) is a good example of a hybrid planner. It is a automated planner that combines generative state-space planning and case-based planning. In generative planning mode (Prodigy 4.0), the system uses search through a space of operator choices in order to build a plan. When in case-based planning mode, the system retrieves the most similar past plans from a case library for use in a given new problem. These plans are reused (replayed) to create a solution for the current goals. The Prodigy 4.0 system (Carbonell, Blythe, Etzioni, Gil, Joseph, Kahn, Knoblock, Minton, Peres, Reilly, Veloso, & Wang, 1992) employs a state-space non-linear planner. It follows a means-ends analysis backward-chaining search procedure that reasons about goals and the operators from its domain theory that are appropriate for achieving such goals. A hierarchy of object classes and a suite of operators and inference rules that change the state of the objects composes the domain theory. A planning problem is represented by an initial state (objects and prepositions about the objects) and a set of goals to achieve. The planning process consists of choosing a goal from a set of pending goals, choosing an operator to achieve the goal, choosing a binding for a given operator, and deciding whether to commit to a plan ordering and to get a new planning state, or to continue expanding unachieved goals. Different choices give rise to different ways of exploring the search space. These choices can be guided by either control rules, by past problem-solving episodes (cases), or by domain-independent heuristics.

Prodigy-Analogy creates plans, interprets and stores planning episodes, and retrieves and reuses multiple past plans that are found similar to new problems. Stored plans are annotated with plan rationale so that, when the plans are retrieved in the future, new decisions can be guided and validated by the past rationale, hence avoiding inefficient search. The derivational-analogy strategy is to derive new solutions based on the decision-making process used in the past, rather than by adapting old solutions created in the past (Carbonell, 1986). Prodigy-Analogy is representative of a hybrid planning system that could be employed in automating maintenance procedure generation in support of T.O. authoring.

3. Conclusions and Recommendations

The authoring and updating of Technical Orders continues to be a time-consuming and labor intensive task that is reflected in very high product life-cycle costs to the Air Force. This remains true in spite of the very significant advances made in the design tools for these products. The advances made in CAD and CAE over the years and the more recent advances in PDM do not yet play an important role in producing and maintaining T.O.s. CAD and CAE can provide much of the input needed for the authoring process and as pointed out by Sanchez et al. (1996), PDM has the potential to address a significant cost area in today's authoring process—accessing relevant engineering data. There is clearly the potential to make significant improvements in the authoring process and these improvements can lead directly to significant cost savings to the Air Force.

Perhaps the most important problem is simply that the authoring process for Technical Orders is, at best, the stepchild of the design process. Dependent as the authoring process is on the PDM/CAD/CAE environment and CAD and CAE data, the authoring process is not an integral part of the design process or the design system. This will certainly change over time and lead to important improvements in the authoring process. As the authoring process becomes an integral part of the design process, the question then becomes: How can an “outside” research and development program be used to enhance a process and a system largely internal to the manufacturers for Air Force equipment?

These observations lead to the first recommendation: The authoring environment for Technical Orders should be developed as a system component within the larger PDM/CAD/CAE system design framework. The authoring environment should, through system PDM capabilities, provide ready access to the CAD and CAE data necessary to support the authoring effort. Given the size and complexity of the authoring task, its close ties to the design process, its dependence on design data, and the industry's movement to fully electronic design processes, it is surprising that such limited progress has been made toward this goal. Since we see this integration as an inevitable consequence of current PDM efforts, recommendations for future research efforts should assume an authoring capability within the larger design framework as a baseline.

The assumption of an authoring capability within an integrated PDM/CAD/CAE environment places some constraints on the form that a research program to support the authoring process should take. The research program should focus on capabilities that can be expected to be integrated into what will be an existing authoring framework, rather than directed toward the design or development of the framework itself. This is consistent with the severe constraints on research budgets and the concern that the research effort address the points of maximum potential leverage. If there is a solid effort underway elsewhere that is addressing a problem, the research program should not overlap that effort. If the research agenda item can yield interesting results, but has little likelihood of being integrating into a large-scale PDM/CAD/CAE environment, then it is not the right agenda item to address.

Working within this framework, we have identified three capabilities (see Four broad technology areas are the basis for developing these AMI authoring capabilities:

computational linguistics, plan representation, automated planning, and human figure modeling. Within computational linguistics we include linguistic analysis, natural language understanding, and text generation. In the planning area, we recommend the development of a hybrid planner based on case-based and generative planning. Technical Order procedures, represented as plans, form the primary data on which each of these technologies operate. CAD and CAE data are essential inputs to support each of these capabilities, but the PDM processes by which these data are obtained is not central to the recommended research effort. In the course of this study, we developed prototype modules and ran test cases to better understand how these technologies might support the development of each AMI system capability. The next logical step in the research agenda is to pursue the development of these AMI capabilities based on these technologies.

Table 6) Shows a list of technologies to improve authoring of Technical Orders. The first is the capability to automatically propose a procedure for a given task that meets the T.O. author's requirements as closely as possible, the second is to use human figure modeling as part of the Technical Order itself and as a tool for the author to use in validating a procedure, and the third is the capability to produce the textual material required for a Technical Order. Each of these capabilities can be developed as a component within the PDM/CAD/CAE framework and each has the potential to perform as an important accelerator in the authoring process.

Four broad technology areas are the basis for developing these AMI authoring capabilities: computational linguistics, plan representation, automated planning, and human figure modeling. Within computational linguistics we include linguistic analysis, natural language understanding, and text generation. In the planning area, we recommend the development of a hybrid planner based on case-based and generative planning. Technical Order procedures, represented as plans, form the primary data on which each of these technologies operate. CAD and CAE data are essential inputs to support each of these capabilities, but the PDM processes by which these data are obtained is not central to the recommended research effort. In the course of this study, we developed prototype modules and ran test cases to better understand how these technologies might support the development of each AMI system capability. The next logical step in the research agenda is to pursue the development of these AMI capabilities based on these technologies.

Table 6. Recommended AMI Capabilities and Supporting Technologies

1. Generate proposed procedure at author's request	
Supporting Technologies:	
<ul style="list-style-type: none"> • data-mining and knowledge acquisition via linguistic analysis and natural language understanding • case-based and generative planning 	
2. Provide animated procedure execution	
Supporting Technologies:	
<ul style="list-style-type: none"> • human figure modeling • linguistic analysis 	
3. Generate textual material for procedure	
Supporting Technologies:	

-
- linguistic analysis and natural language text generation
-

The planner will be called upon to propose procedures that will be as complete and as accurate as possible. The goal is to provide procedures that will require few adjustments, leaving validation as the author's most critical task. The case-based planner is the critical technology to bring the planning capability on-line as soon as possible. With that research activity underway, the effort to develop and integrate the generative planner should be initiated. The data-mining capability is essential to building the case-base for supporting the planning process. The linguistic analysis and natural language understanding necessary to provide the data-mining capability should be pursued. The research effort should address the development and integration of the data-mining and planning capabilities.

Text generation has the potential to provide the AMI author with much of the textual material necessary for a Technical Order. The linguistic analysis of existing Technical Orders that is used to support the language understanding effort should be pursued to support the research and development for the text generator.

Human figure modeling can play an important role in validating Technical Order procedures and can be developed to provide material for the Technical Orders themselves. The research effort underway to provide human figure modeling, based on the textual description of procedures, should continue so that this potential might be realized.

The new AMI capabilities recommended in this study can be brought on-line as the new PDM/CAD/CAE system design environments become available. Results based on this research agenda can be achieved in the short and intermediate term. Incremental improvements will also be possible. As PDM provides better access to CAD and CAE data, each of the AMI capabilities can be further improved. We would expect results to be useful to the organizations producing T.O.s within the next five years. Air Force investment in the selected AMI authoring capabilities and the technologies needed to support these capabilities will lead to reduced costs in development and updating of Technical Orders.

4. Glossary

Case Based Planning: A planning technique where past problem solving situations and solutions are stored as cases in a memory (the case-base), indexed, and reused to solve similar problems. Also known as "planning from experience".

Chart Parser: A parsing system that doesn't commit to a single possible path while building a parse, but explores every possible path in parallel.

Data Mining: The extraction of hidden predictive information from large databases.

Experiential Knowledge: Knowledge that people gain from experience and practice, common sense.

Generative Planning: A planning technique which creates plans for specific circumstances by elaborating general schemas and specific knowledge about a domain. The general schema is viewed a set of goals; planning consists of recursively decomposing these goals into subgoals and proposing actions that accomplish the goals.

Implicit Information: Knowledge that isn't explicitly stated in a text, either because it is assumed the reader has this knowledge, or the author has determined this knowledge isn't essential for the reader's comprehension of the text.

Knowledge Acquisition: The task of identifying sources of domain relevant knowledge, studying those sources to learn the knowledge, and developing appropriate knowledge database formalisms that can accurately represent this knowledge.

Language Understanding: The parsing of input texts into a knowledge representation that can be used by other software modules.

Linguistic Analysis: The careful study of a body of text with the goal of identifying generalizations exhibited in the text, important stylistic details such as level of detail in presentation, and the organization of the text.

Ontology: The description of an entity in terms of its intrinsic properties and its relationships to other entities. This commonly takes the form of a taxonomy, a classification showing the similarities between objects arising from the inheritance of properties. Another important relationship captures the decomposition of entities into their constituent parts; i.e., part-whole relation.

Plan Representation: A plan, such as a maintenance procedure, in computer-based form that might be generated from natural language text and modified by an automated planner or by a person using a plan editing environment.

Semantic Category: A category designed to reflect the meaning of, or knowledge contained in, a portion of text.

Semantic Parsing: Parsing that seeks to label portions of the text with semantic labels that reflect the meaning of the parsed text.

Semantic Phrase Structure Grammar: A grammar whose rules define the decomposition of semantic categories into other semantic categories, to be used for semantic parsing.

Syntactic Category: A category defined to reflect the structure of a portion of text.

Syntactic Parsing: Parsing that seeks to label portions of the text with syntactic labels that reflect the sentence structure of the parsed text.

Syntactic Phrase Structure Grammar: A grammar whose rules define the decomposition of syntactic categories into other syntactic categories, to be used for syntactic parsing.

5. References

Badler, N., Webber B., Palmer, M., Noma, T., Stone, M., Rosenzweig, J., Chopra, S., Stanley, K., Dang, H., Bindiganavale, R., Chi, D., Bourne, J., & DiEugenio, B. (1997). *Final report to Air Force HRGA regarding feasibility of natural language*

-
- text generation from task networks for use in automatic generation of technical orders from DEPTH simulations*. Philadelphia, PA: University of Pennsylvania.
- Burstein, M. (1994). Case age: Selecting the best exemplars for plausible reasoning using distance in time and space. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, Erlbaum Associates.
- Carbonell, J. G. (1986). Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. In R. Michalski, J. Carbonell & T. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach*, Vol. 2, 371-392.
- Carbonell, J. G., Blythe, J., Etzioni, O., Gil, Y., Joseph, R., Kahn, D., Knoblock, C., Minton, S., Peres, A., Reilly, S., Veloso, M. M., & Wang, X. (1992). *PRODIGY 4.0: The manual and tutorial*, Department of Computer Science, Carnegie Mellon University, Tech. Rep. CMU-CS-92-150.
- Deutsch, S. E., & Adams, M. J. (1995). The operator-model architecture and its psychological framework. *6th IFAC Symposium on Man-Machine Systems*. MIT, Cambridge, MA.
- Deutsch, S. E., Adams, M. J., Abrett, G. A., Cramer, N. L. & Feehrer, C. E. (1993). *Research, Development, Training, and Evaluation (RDT&E) Support: Operator Model Architecture (OMAR) software functional specification* (AL/HR-TP-1993-0027). Wright-Patterson AFB, OH: Armstrong Laboratory, Logistics Research Division.
- Deutsch, S. E., MacMillan, J., Cramer, N., & Chopra, S. (1997). *Operability Model Architecture demonstration final report* (AL/HR-TR-1996-0161). Wright-Patterson AFB, OH: Armstrong Laboratory, Logistics Research Division.
- Freeman, B. (1997). *OMAR user/programmer manual*. BBN Technical Report No. 8181. Cambridge, MA: BBN Technologies.
- Goel, A. & Chandrasekaran, B (1989). Case-based design: A task analysis. In K. Hammond (Ed.), *Proceedings of the DARPA Case-Based Reasoning Workshop*, 100-109.
- Hammond, K. J. (1986). CHEF: A model of case-based planning. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, 267-271.
- Kolodner, J. (1993). *Case-based reasoning*. San Mateo, Ca.: Morgan Kaufmann Publishers.
- Koton, P. (1988). Reasoning about evidence in causal explanations. In *Proceedings of AAAI-88*, 256-261.
- Leake, D. B. (1997). *Case-based reasoning, experiences, lessons, and future directions*. AAAI Press, Menlo Park, CA.
- Mark, W, Simoudis, E., & Hinkle, D. (1997). Case-based reasoning: Expectations and results. In D. B. Leake (Ed.), *Case-based reasoning, experiences, lessons, and future directions*, 269-294.

- McDonald, D. (1992). An efficient chart-based algorithm for partial parsing of unrestricted texts. *Proceedings of the 3rd Conference on Applied NLP*.
- Mulvehill, A. M. (1995). Reusing force deployment plans. *AAAI 1995 Fall Symposium on Adaptation of Knowledge for Reuse*, Cambridge, MA.
- Mulvehill, A. M., & Christopher, D. G. (1991). *Computer care center project - final year end report*, MITRE MTR-11104, The MITRE Corporation, Bedford, MA 01730.
- Navinchandra, D. (1988). Case-based reasoning in CYCLOPS, a design problem solver. In J. Kolodner (Ed.), *Proceedings of the DARPA Case-Based Reasoning Workshop*, 286-301.
- Nicholson, J. R., & Mulvehill, A. M. (1990). Expert knowledge techniques applied to the analysis of electric field mill data, *16th Conference on Severe Local Storms, Conference on Atmospheric Electricity*, Oct 22 - 26, 1990, Kananaskis Park, Alberta Canada.
- Riesbeck, C. L. & Shank, R. C. (1989). *Inside case-based reasoning*, Hillsdale, NJ: Lawrence Erlbaum Associates.
- Sanchez, E., Winning, S., & Boyle, E. S. (1997). *Automated support for maintenance technical manuals* (AL/HR-TP-1997-0051). Wright-Patterson AFB, OH: Armstrong Laboratory, Logistics Research Division.
- Shank, R. C. (1997). Goal-based scenarios: Case-based reasoning meets learning by doing. In D. B. Leake (Ed.), *Case-based reasoning, experiences, lessons, and future directions*, 295-349.
- Stroulia, E., Shankar, M., Goel, A., & Penberthy, L. (1992). A model-based approach to blame assignment in design. *Artificial Intelligence in Design*. In J. Gero (Ed.), 519-537.
- Swartout, W. R. (1981). *Producing explanations and justifications of expert consulting programs*. Technical Report MIT/LCS/TR-251. Cambridge, MA: Massachusetts Institute of Technology Laboratory for Computer Science.
- Sycara, K. P., Guttal, R., Koning, J., Narasimhan, S., & Navinchandra, D. (1991). CADET: A case-based synthesis tool for engineering design. *International Journal of Expert Systems*, 4(2), 157-188.
- Veloso, M. (1994). *Planning and learning by analogical reasoning*. Springer-Verlag.
- Voss, G., Grather, & Schmidt (1997). *Reasoning with complex cases*. In Press.
- Winograd, T. (1983). *Language as a Cognitive Process, Vol. 1: Syntax*, Reading, Mass.: Addison-Wesley Publishing Company.
- Zimmerman, D. L., Green, C., Gunning, D., Worrall, & Dimock, B. (1993). *Report on the workshops: Automated generation of electronic technical manuals*. (AL/HR-TP-1993-0036). Wright-Patterson AFB, OH: Air Force Materiel Command.

THIS PAGE INTENTIONALLY LEFT BLANK